Università degli Studi di Milano - Bicocca

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di Laurea Magistrale in Informatica

Video Restoration using Convolutional Neural Networks

Relatore: Prof. Simone Bianco

Co-relatore: Dott. Marco Buzzelli

Tesi di Laurea Magistrale di:
Claudio Rota
Matricola 816050

Anno Accademico 2020-2021

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the last decade, the number of multimedia contents produced every day has considerably increased due to the growing diffusion of digital devices, such as digital cameras and smartphones. Nowadays, images and videos are widely used in different fields, ranging from social media to self-driving cars. Although modern cameras are able to capture high-quality images and videos, there are some cases in which the quality of these contents is significantly reduced. When an image is captured in poor light conditions, free electrons within the acquisition device may corrupt the digital signal, introducing noise. When capturing dynamic scenes, camera movements during the acquisition process and poor focus selection may produce blurred contents. In addition, compression algorithms are often used to reduce memory occupation of both images and videos, but they also introduce visible artifacts such as blocking, contouring and ringing effects. In all these examples, the quality of images and videos is reduced because of artifacts damaging their contents, causing problems to both user experience and many computer vision tasks.

Convolutional neural networks (CNNs) have recently shown incredible results in many computer vision tasks, from image classification to video segmentation, outperforming traditional methods. Such success has led more and more researchers and industries to invest time and money in developing deep neural networks to address different real-world problems.

For these reasons, several deep learning approaches to restore the quality of degraded images and videos have been introduced in the literature under the name of image and video restoration methods.

Video restoration aims to recover the clean video sequence from its degraded version. Based on the degradation operators affecting the video sequence, such as noise or blur, there are different video restoration tasks: video denoising aims to remove noise from video frames; video deblurring has the purpose of restoring blurred contents; video compression artifact reduction has the objective of recovering the original frames removing artifacts introduced by compression algorithms. Despite many methods to restore videos affected by different artifacts have been proposed in the literature, the vast majority of them are designed to deal with a specific distortion type. Such methods produce excellent results on videos affected by the considered artifacts, but they may produce unacceptable results when multiple artifacts are present. For instance, existing approaches for video compression artifact reduction do not consider the case of noisy videos, in which compression algorithms produce artifacts that are more difficult to remove. Therefore, having a single framework able to restore videos even when they are affected by multiple artifacts simultaneously can be very useful.

Possible applications are in mobile devices, such as smartphones, frequently used to capture a wide variety of scenes in different contexts, in videoconferencing software, in which video frames are usually compressed to reduce bandwidth, and in surveillance cameras, in which videos are often captured in poor light conditions.

1.1 Content of the thesis

This thesis sets the goal of developing a deep neural network able to restore multi-distorted videos, that is, videos corrupted by multiple degradation operators. To this end, the state of the art of video restoration has been deeply investigated, and a promising approach for video denoising has been selected as baseline model and further studied to extract its best characteristics, allowing the design of Multi-distorted Video Restoration Network, called MdVRNet.

MdVRNet is a two-stage restoration network that progressively aligns adjacent frames, allowing to extract both spatial and temporal information from the target frame and its adjacent ones. The first restoration stage pays more attention to single pixel restoration, due to its limited temporal information, while the second restoration stage pays more attention to restoring local areas, as it has a more complete vision of the entire scene. MdVRNet exploits an original distortion parameter estimation module specifically devised to obtain information about degradation operators affecting the video sequence to make the restoration process more robust. The proposed framework uses a novel multi-scale restoration block to extract features at different scales using two parallel streams: the full-resolution stream learns fine pixel dependencies for finer detail reconstruction, while the low-resolution stream learns coarse pixel dependencies to make the most of the semantic in local areas. In addition, it uses an attention mechanism to weight the features extracted by the two parallel streams according to their importance in reconstructing the target frame.

An extensive experimentation has been carried out with different purposes, including but not limited to assessing the effectiveness of the proposed MdVRNet in restoring videos affected by multiple distortions, noise and compression artifacts to be precise. From the results, it emerged that providing MdVRNet with information related to degradation operators using the parameter estimation module integrated into the framework allows to increase the restoration performance. Moreover, also the novel multi-scale restoration block helps improve the results, showing the usefulness of having two parallel branches working at different scales and a mechanism to weight the extracted features according to their importance in reconstructing the target

frame.
The rest of this thesis is structured as follows: Chapter 2 introduces the most common degradation operators that can reduce the quality of images and videos; Chapter 3 presents several state-of-the-art approaches based on deep learning techniques for both image and video restoration, as well as for contrast enhancement, reporting the architectures they use and their main ideas; Chapter 4 describes the methodological approach for video restoration followed in this thesis that allowed to design the MdVRNet framework, starting from the analysis of characteristics and components of the video restoration methods presented in the previous chapter and concluding with a detailed description of the method proposed to restore multi-distorted videos; Chapter 5 reports all the experiments carried out to evaluate different aspects of both the baseline model and the proposed MdVRNet, mentioning interesting considerations based on the obtained results; Chapter 6 concludes the thesis by briefly summarizing all the contents presented in the previous chapters and mentioning possible future work.

Chapter 2

Degradation operators

This chapter introduces the most common degradation operators that can reduce the quality of images and videos, mentioning the possible causes and showing some examples. Such decrease in quality can negatively impact user experience or make some computer vision tasks fail when the processed multimedia contents are expected to be of high quality.

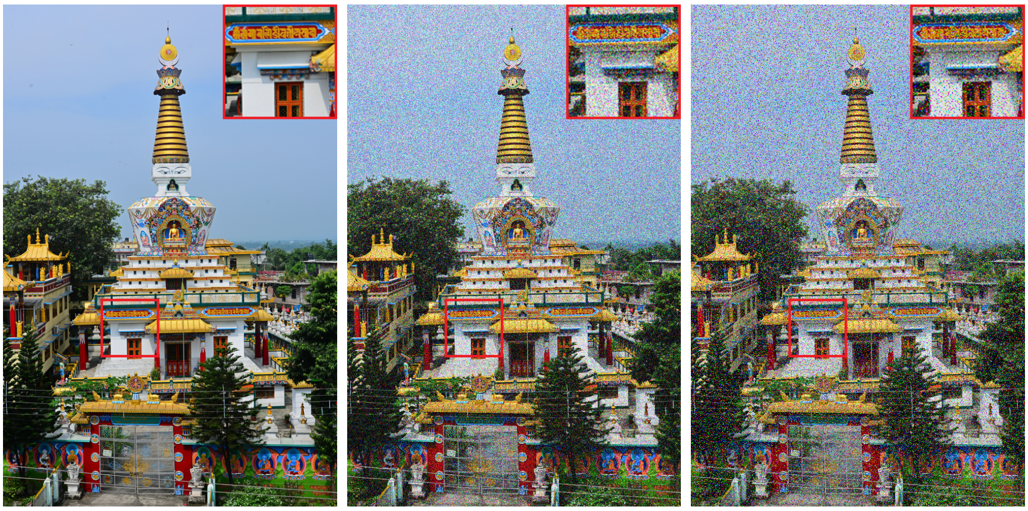
More in detail, Section 2.1 presents the problem of noise, Section 2.2 addresses the problem of blur, Section 2.3 describes the problem of artifacts introduced by compression algorithms and, finally, Section 2.4 shows how the artifacts change when multiple distortions are present, justifying why a method able to handle artifacts introduced by multiple degradation operators is needed.

2.1 Noise

When images are captured by camera sensors, the output signal of some of these sensors may be corrupted by noise. There are different sources of noise in images. For instance, free electrons within acquisition devices can alter the digital signal, and such alteration becomes more visible when the signal entering the sensors is weak, i.e. there is not enough light. This is why noise is more present in low-light images rather than in bright ones.

Noise can be either additive or multiplicative. Let X be the original image, N the additive noise and I the noisy observation, the noise degradation model can be formally defined as $I = X + N$. There are different types of additive noise, additive white Gaussian noise (AWGN) and impulsive noise are the most commonly treated ones. Additive white Gaussian noise manifests itself in random perturbations of pixel values depending on the noise intensity. It is defined by two parameters, the mean μ and the standard deviation σ. Usually, the mean is considered as zero, while the value of σ determines the intensity of the noise: the higher the value, the higher the noise intensity. Impulsive noise, commonly called salt-and-pepper noise, manifests itself in random pixels changed to 0 (salt) or 255 (pepper). It is defined by the density parameter ρ, which represents the percentage of noisy pixels.

Examples of images affected by additive white Gaussian noise and impulsive noise are shown in Figure 2.1.



(a) Original

(b) Additive white Gaussian noise with $\mu = 0$ and $\sigma = 50$

(c) Impulsive noise with $\rho = 0.1$

Figure 2.1: Example of images affected by additive white Gaussian noise and impulsive noise

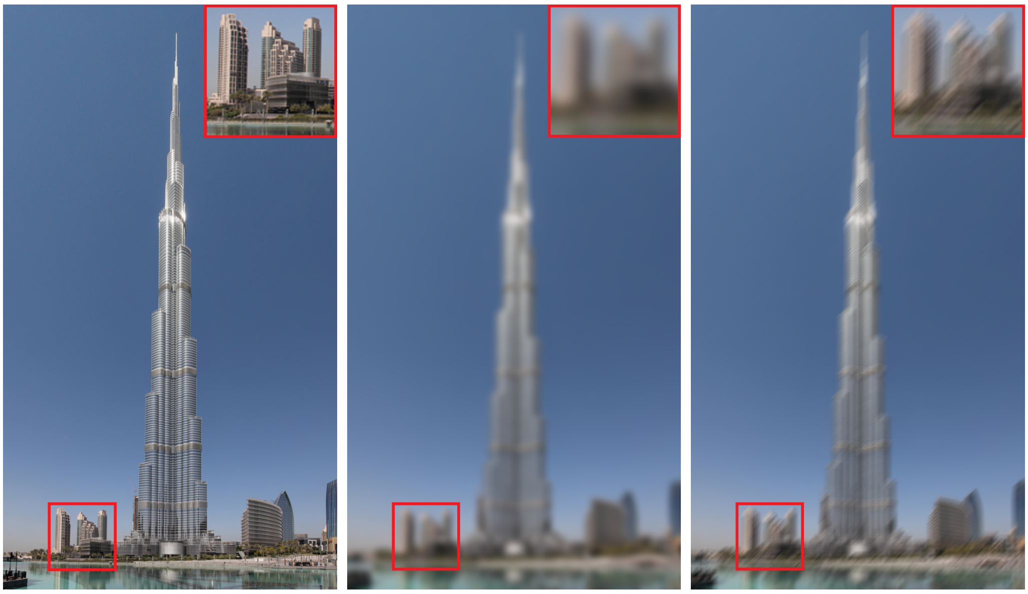
In this thesis work, only additive white Gaussian noise has been considered because it is the most common type of noise several video denoising approaches have been designed for.

2.2 Blur

There are different causes of blurred images, such as atmospheric distortion motions, optical aberrations, motion and so on. Motion blur is one of the most common problem when taking photos. For example, camera shaking and fast object motion can degrade the image quality by producing blurred contents. Moreover, when an image is captured by camera devices, some points are in focus while others may not, thus causing out-of-focus blurring. Out-of-focus blurring is space-invariant in those cases in which the surface of a flat object is parallel to the image plane, but it may not be uniformly distributed, as there are some objects in focus while others are not [1].

Let X be the original image, K a blur kernel and B the blurred observation, the blur degradation model can be formally defined as $B = K \times X$, where $\times$ denotes the 2D convolution operation.

Due to the limited availability of real data for both image and video deblurring, synthetic samples must be generated in order to train deep neural networks. There are different ways to create synthetic data for this task. For example, out-of-focus blur can be simulated by means of Gaussian kernels, which are defined by two



(a) Original

(b) Blurred using a Gaussian kernel with $\sigma = 6$

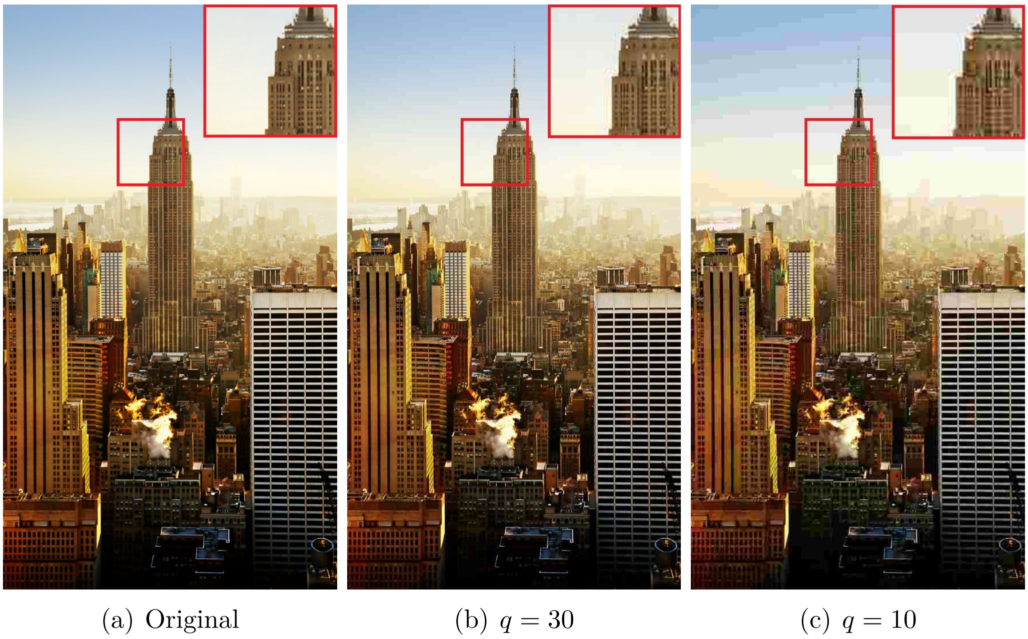
(c) Blurred using a linear motion kernel with $\theta = 45$ and $L = 20$

Figure 2.2: Example of images blurred using a Gaussian kernel and a linear motion blur kernel

parameters: the kernel dimension and the value of the standard deviation σ. It is worth mentioning that the kernel dimension can be derived from the value of σ, hence the latter is the only parameter to specify. To simulate motion blur, it is possible to design either linear or non-linear motion kernels. Linear motion kernels are defined by θ, which represents the blur direction, and L, which represents the blur length. Non-linear kernels are much more complex, requiring specifically designed functions. Figure 2.2 shows an example of the same image blurred with a Gaussian kernel and a linear motion blur kernel.
In this thesis, among the different types of blur degradation, only out-of-focus blur has been considered because it can be easily synthesized using Gaussian kernels.

2.3 Compression

Compression algorithms are fundamental to store images and videos in memory limited devices. There are several compression methods applicable to both videos and images, and they can be mainly divided into two classes: lossless methods and lossy methods. Lossless methods can entirely recover the original image from the compressed one. Instead, lossy methods, such as JPEG, remove high-frequencies from images and do not allow to recover the exact uncompressed image: they are, indeed, lossy. The JPEG compression acts by dividing an image into 8×8 blocks and removing high-frequencies from each block individually by decomposing them with a discrete cosine transform (DCT) based method. The entity of the compression is controlled by a quality factor q in the $[0, 100]$ range: low values mean high compression, high values mean low compression. When q is very low, the vast majority of high-frequencies within each block are removed, producing blocking artifacts.



(a) Original (b) $q = 30$ (c) $q = 10$

Figure 2.3: Example of images compressed using the JPEG algorithm with different values of the quality factor q

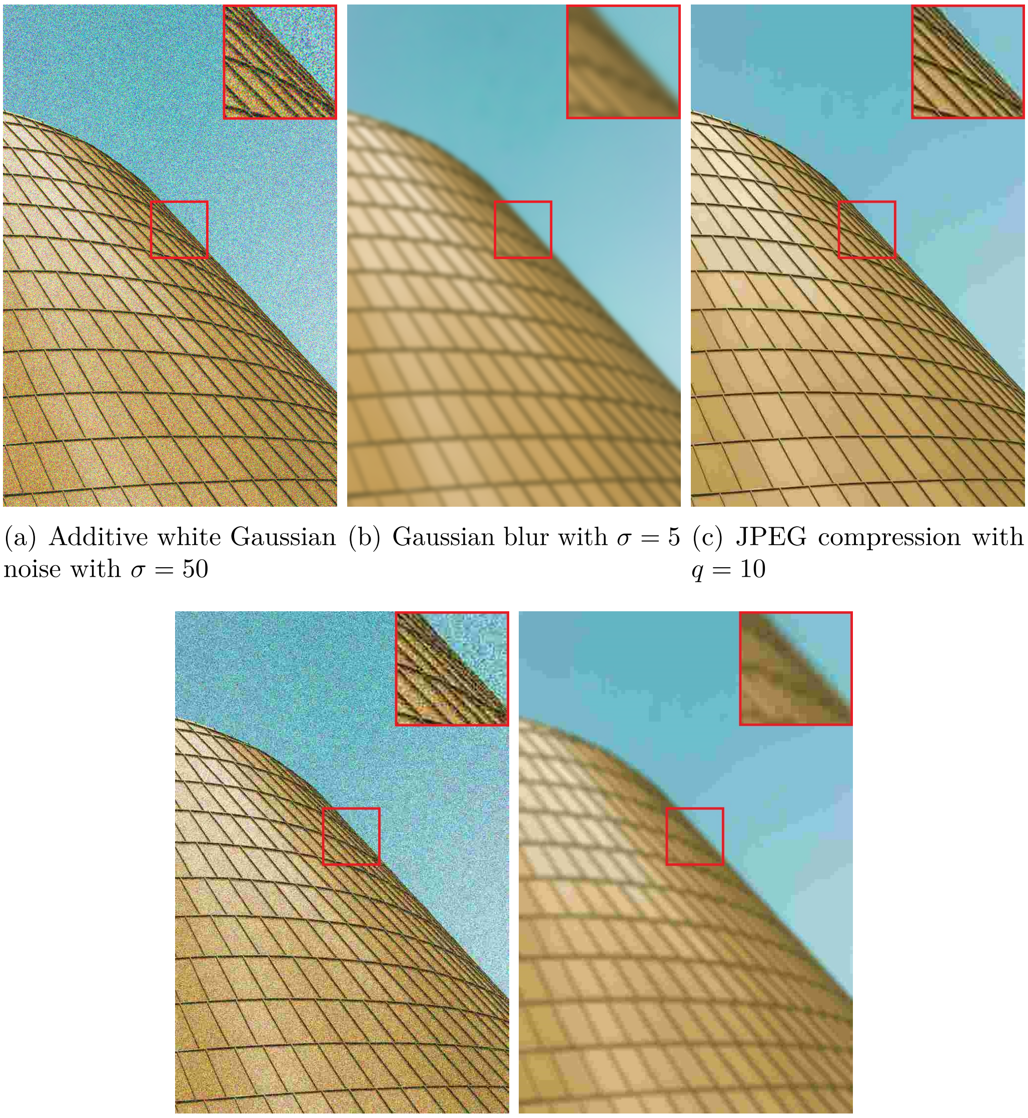
Figure 2.3 shows examples of blocking artifacts arising when an image is compressed with different values of q. It is worth mentioning that the size of the original image is about 800 KB, the size of the image compressed with $q = 30$ is about 56 KB and the size of the image compressed with $q = 10$ is about 28 KB.

Compressing a video sequence with the JPEG algorithm means splitting it into single frames and treating each frame independently. Moreover, some compression methods, such as AVC and HEVC, use inter-frame compression taking advantage of temporal redundancy between adjacent frames to enable higher compression rates, producing quality fluctuation among frames, i.e. some frames are more compressed than others.

This thesis is built on the constraint that all the frames of a video sequence should have the same quality after being compressed. This is done in order to address a worst-case scenario where no cross-frame information is available. Hence, among all compression methods, the JPEG compression has been chosen because it is widely used, lossy and applicable to each frame independently.

2.4 Multiple distortions

As already said, compression algorithms, such as JPEG, are widely used to reduce memory occupation, but applying such algorithms may introduce blocking artifacts. Moreover, compression algorithms are often applied to images already corrupted by other distortions, such as noise or blur, producing new types of artifacts.



(a) Additive white Gaussian noise with $\sigma = 50$

(b) Gaussian blur with $\sigma = 5$

(c) JPEG compression with $q = 10$

(d) Additive white Gaussian noise with $\sigma = 50$ and JPEG compression with $q = 10$

(e) Gaussian blur with $\sigma = 5$ and JPEG compression with $q = 10$

Figure 2.4: Example of images affected by additive white Gaussian noise, Gaussian blur, JPEG artifacts and combination of them

One may think that a possible solution for dealing with videos affected by multiple artifacts is that of restoring them by placing a sequence of different distortion-specific restoration methods in cascade. For instance, a noisy and compressed video can be restored by applying a compression artifact reduction method and a denoising method in sequence, as compression artifacts are usually introduced after noise and they should be removed first. However, the shape and the distribution of the artifacts introduced by noise or compression algorithms are different from the ones of artifacts that applying a compression method on a noisy video sequence produces, thus restoring it using two artifact-specific models in cascade may not produce good results.

Figure 2.4 shows five versions of an image corrupted by additive white Gaussian noise, Gaussian blur, JPEG compression artifacts and combinations of them. In this latter case, the JPEG algorithm is applied either to the noisy or the blurred image. As shown, the artifacts in Figure 2.4(d) and 2.4(e), representing distortion combinations, are more complex than the artifacts introduced by single distortions, making the restoration problem more difficult to solve. This is why a restoration approach able to restore videos even in the presence of multiple artifacts is needed. It is worthwhile to mention that, having three different distortion types, several distortion combinations are possible. However, some of them have better application than others in real-world scenarios. According to the image acquisition pipeline, blur is the first distortion introduced, followed by noise and finally by compression artifacts. Following this observation, in this thesis only the combinations noise/compression and out-of-focus blur/compression have been investigated.

Chapter 3

State of the art

This chapter presents the state of the art of video restoration. In particular, several approaches based on deep learning techniques are detailed under different aspects, such as the key ideas and the architectures used.
It starts presenting some state-of-the-art approaches to single image restoration, discussed in Section 3.1, and then it extends to the video domain in Section 3.2. Finally, Section 3.3 briefly presents some state-of-the-art approaches to image and video contrast enhancement, a task highly related to image and video restoration, that are worth analyzing to gather more information about what exists in the literature to enhance images and videos using deep learning techniques.
The following approaches are reported in order of the publication date of their corresponding publications.

3.1 Image restoration

Before presenting the state-of-the-art approaches for video restoration, it is better to introduce some works related to single images, as some video restoration approaches are based on them.
Starting from 2014, due to the success of convolutional neural networks (CNNs) in image classification, many researches have been designing deep learning frameworks to deal with image restoration tasks, which include super resolution, deblurring, denoising as well as compression artifact removal.

3.1.1 SRCNN

The first work that applied deep learning techniques to restore single images dates back to 2014, when Dong et al. [2] proposed to use deep learning for single image super resolution. They took inspiration from a traditional approach for image super resolution [3] and showed that the same operations could be performed using a CNN. Hence, they designed a simple architecture for image super resolution, taking

the low-resolution image as input and producing the corresponding high-resolution image as output.
The proposed architecture, named SRCNN, is shown in Figure 3.1. It consists of three convolutional layers: the first layer extracts patches from the low-resolution input image and maps them into a set of feature maps; the second layer applies a non-linear mapping to map the input feature maps into another set of feature maps, which conceptually are the representation of the extracted patches in a high-resolution space; the third layer, finally, aggregates these high-resolution patches to generate the final high-resolution image.

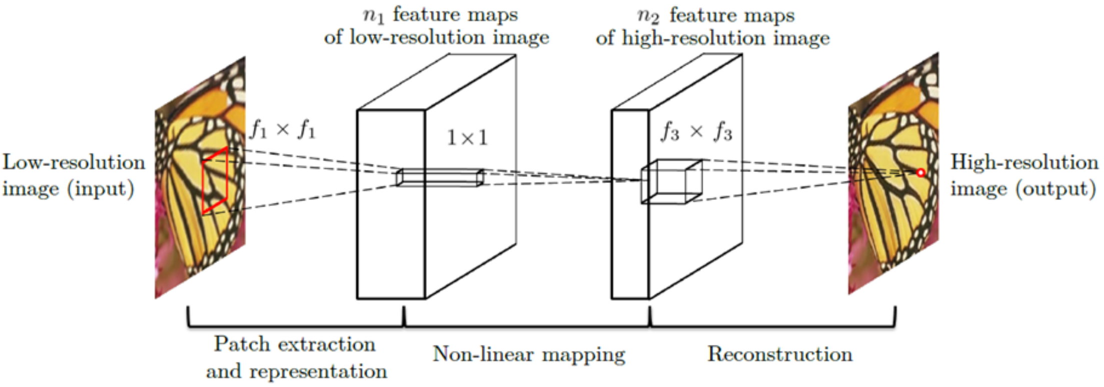


Figure 3.1: Architecture of SRCNN for image super resolution [2]

Before feeding the image to the network, it is upscaled using bicubic interpolation so that the input spatial dimension is the same as the desired output one. Note that the proposed network does not perform any downsampling, keeping the spatial dimension of the feature maps fixed.

3.1.2 ARCNN

In 2015, Dong et al. [4] extended SRCNN [2] proposing a new architecture, named ARCNN, capable of reducing compression artifacts. They noticed that, when dealing with compression artifacts, the feature maps extracted from the first layer of SRCNN [2] were too noisy, hence inadequate for an accurate mapping. To solve this problem, the authors inserted an additional convolutional layer after the first convolutional layer of SRCNN [2] to perform feature enhancement. Intuitively, such layer enhances the features extracted from the input image, improving the mapping accuracy.
An overview of ARCNN is given by Figure 3.2. As shown, the architecture of ARCNN is very similar to SRCNN [2] except for the additional convolutional layer for feature enhancement.
Besides introducing ARCNN, they noticed that, when the compression factor was quite high, the architecture they designed had convergence problems. To address such problem, they experimented the “easy-hard transfer” for weight initialization:

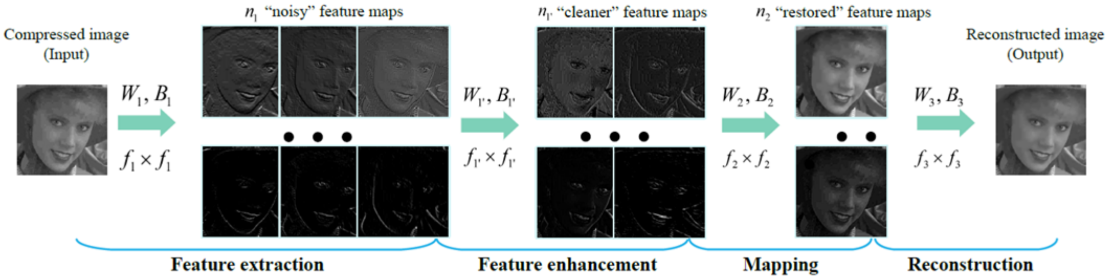


Figure 3.2: Architecture of ARCNN to remove compression artifacts from images [4]

instead of training the network from scratch when the compression factor was high, they initialized the network with the weights obtained from networks trained with lower compression factors. This approach effectively solved the convergence problem.

3.1.3 DNCNN

In 2016, Zhang et al. [5] proposed to use a CNN for single image denoising. While previous works on single image denoising tried to recover the clean image from its noisy counterpart, they exploited residual learning [6] to train a CNN, called DNCNN, to learn the noise image that added to the clear image produces the input noisy image. Therefore, as they assumed the noise to be additive, the clean image could be easily obtained by subtracting the predicted noise image from the input noisy image. To design their network, they adapted the VGG architecture [7] to make it able to handle image denoising.
The DNCNN architecture is reported in Figure 3.3. Depending on the noise level, DNCNN can be more or less deep. Indeed, when the noise level is high, the deeper network performs better than the shallower one.

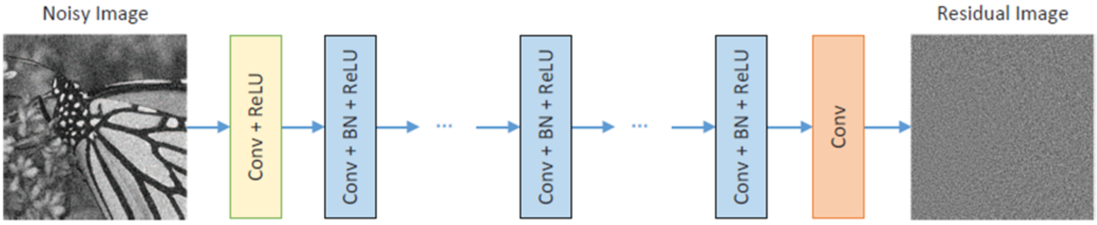


Figure 3.3: Architecture of DNCNN for image denoising [5]

In addition to the use of residual learning [6], the authors also used batch normalization [8] both to speed up the training process and to boost the denoising performance. Zhang et al. [5] also demonstrated that their network could be trained to tackle other image restoration tasks, such as blind Gaussian denoising, JPEG artifact removal and image super resolution.

3.1.4 ESPCN

Shi et al. [9] improved SRCNN [2] by introducing a new way to perform upscaling, in 2016. In particular, they stated that SRCNN [2] was computationally too expensive because it increased the spatial resolution of images to work directly in the full-resolution space. They also stated that the use of bicubic interpolation to perform upscaling may not be suitable because it does not bring additional information to solve the upscaling problem. Therefore, to address these drawbacks, Shi et al. [9] proposed to increase the spatial resolution at the end of the network using a novel sub-pixel convolutional layer, creating a new network architecture called ESPCN.

As it is possible to see in Figure 3.4, the sub-pixel convolutional layer, also called Pixel Shuffle layer, is applied at the end of the network, allowing the latter to perform most of the work in the low-resolution space. This significantly reduces the number of operations required to produce the final output. Moreover, employing a sub-pixel convolutional layer, whose weights are learned during the training process, allows ESPCN to learn the optimal upscaling operation enabling it to produce better results.

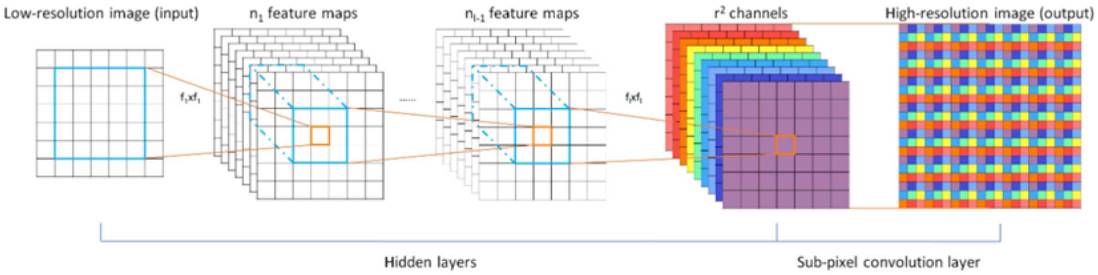


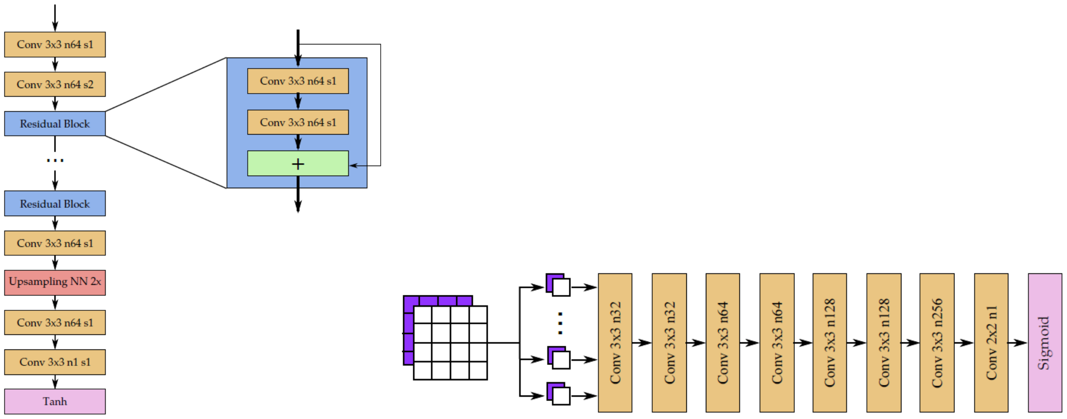
Figure 3.4: Architecture of ESPCN for image super resolution [9]

3.1.5 DGCAR

Galteri et al. [10], in 2017, were the first to propose the use of Generative Adversarial Networks [11] (GANs) to address the problem of compression artifact removal.

The GAN they designed, named DGCAR, is composed of a generator, which aims to generate clean images starting from images affected by compression artifacts, and a discriminator, which aims to improve the generator performance. Figure 3.5(a) shows the used generator, while Figure 3.5(b) shows the used discriminator. Instead of using MSE as loss function, which is widely employed in many deep learning tasks, they designed a specific loss based on SSIM [12], allowing to perceptually improve the results.

In addition, instead of feeding the entire image to the discriminator, they just fed image sub-patches. This because, when dealing with compression artifacts, it is not important to see the entire image, as compression algorithms usually decompose images into patches and artifacts are typically created within them.



(a) Generator

(b) Discriminator

Figure 3.5: Generator and discriminator composing the DGCAR architecture to remove compression artifacts from images [10]

3.1.6 DeepDeblur

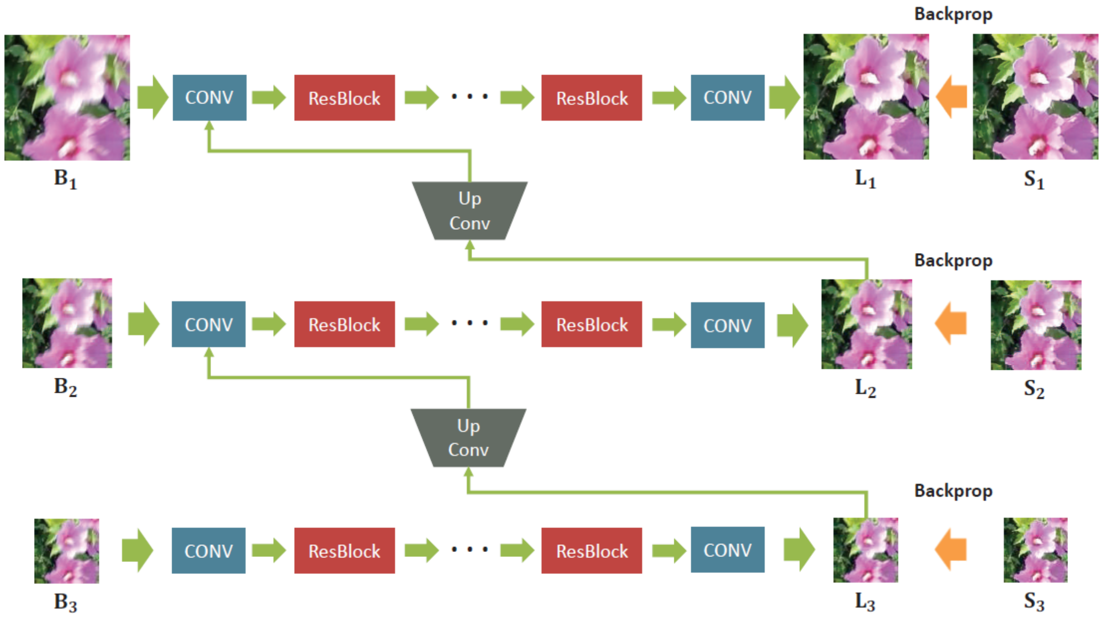


Figure 3.6: Architecture of DeepDeblur for image deblurring [13]

In 2017, Nah et al. [13] proposed a novel architecture to tackle the image deblurring problem. Previous works on image deblurring estimate blur kernels and then apply them to recover the latent clean image. However, blur kernel estimation can be difficult in some situations and, when such estimation is not correct, using these kernels to restore blurred images may cause visible artifacts. To avoid this problem, Nah et al. [13] adopted a kernel-free model and designed a multi-scale architecture, named DeepDeblur, to mimic conventional coarse-to-fine optimization methods.

The proposed framework is shown in Figure 3.6. DeepDeblur uses three scales, each of which ideally produces the sharp image at that scale. Such sharp image is upscaled and concatenated with the blurred image at the higher scale. Instead of upscaling the outputs of the lower scales using non-learnable techniques, such as bilinear or bicubic interpolation, they adopted upconvolutions. Indeed, since blurred and sharp images share the same low frequencies, the use of upconvolutions may help remove redundancy.
They also designed a specific loss to train their network in such a way that the output at each scale is as similar as possible to the ground truth at that scale. Moreover, to improve the deblurring performance, they trained DeepDeblur in an adversarial way by designing an appropriate discriminator.

3.2 Video restoration

The main difference between image and video restoration is that, when dealing with videos, both spatial and temporal information must be taken into account to produce optimal results. In fact, one may see video restoration as multiple image restoration, in which every frame is restored independently. However, this method cannot exploit temporal information between frames, which is important to produce high-quality results and avoid flickering artifacts.
The following approaches tackle different video restoration tasks, that is, temporal frame interpolation, super resolution, compression artifact reduction, denoising and deblurring. Some works address just a single task, while others address more tasks. Nevertheless, it is important to notice that, except for one method, none of them deals with multiple artifacts jointly.

3.2.1 TOFlow

The work carried out by Xue et al. [14] in 2017 represents one of the first attempts to apply image restoration tasks to videos using deep learning. They designed a framework to deal with four restoration tasks, that is, temporal frame interpolation, super resolution, denoising and compression artifact removal.
Their framework, TOFlow, is shown in Figure 3.7. It consists of three modules: flow estimation, transformation and image processing. In brief, using its flow estimation module, TOFlow estimates pixel motion between the target frame and each of its neighboring frames. Then, the transformation module warps each neighboring frame to the target one using the estimated pixel motion. Finally, the information contained in the warped frames is used to restore the target one using the image processing module. The flow estimation module estimates the movement between the target frame and a frame of its neighborhood. To do this, TOFlow exploits SpyNet [15], a deep neural network for optical flow estimation that uses a coarse-to-fine spatial pyramid structure to handle large motion. The transformation module makes use of a spatial transformer network [16] (STN), which is a differentiable bilinear interpolation layer, to warp all the neighboring

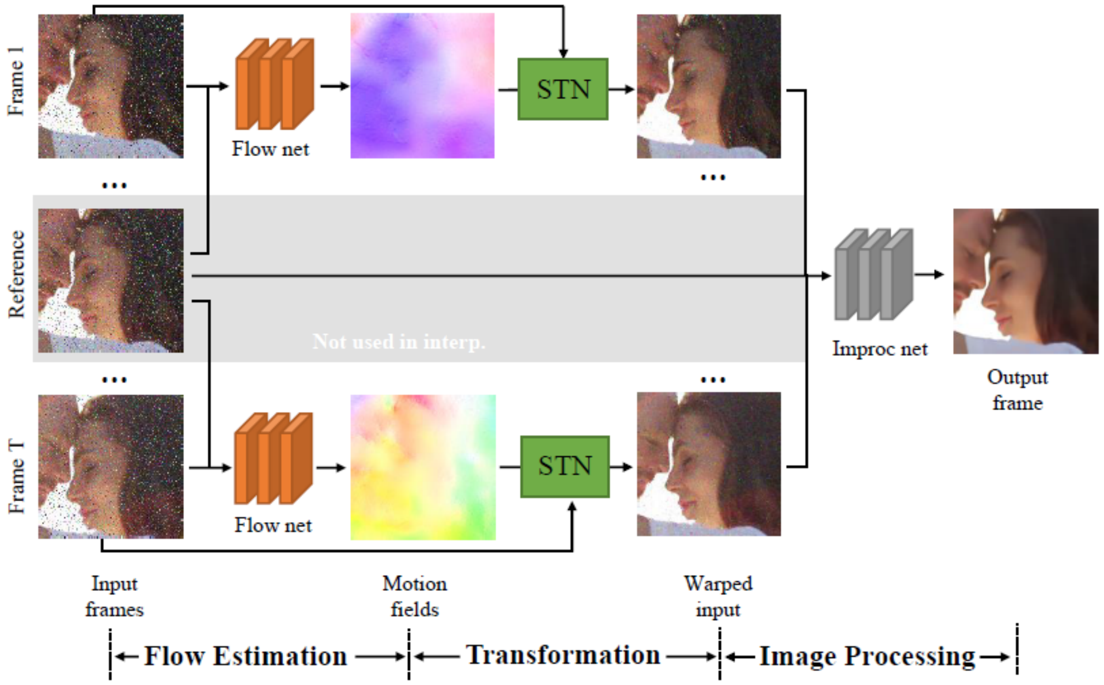


Figure 3.7: TOFlow framework for video temporal frame interpolation, denoising, super resolution and compression artifact removal [14]

frames to the target one, i.e. each frame is transformed into the viewpoint of the target one. The image processing module restores the target frame using both the spatial and temporal information extracted from all the warped frames.

An important aspect of TOFlow is that it can be trained end-to-end, allowing to propagate the gradient from the image processing module back to the flow estimation module.

3.2.2 DeBlurNet

In 2017, Su et al. [17] proposed DeBlurNet to address blur produced by camera shaking. Unlike TOFlow [14], DeBlurNet is able to exploit spatial and temporal information coming from multiple frames to restore the target one without using specific modules for explicit motion estimation and compensation.

Figure 3.8 shows the architecture of DeBlurNet. It is an encoder-decoder architecture that takes a stack of consecutive frames as input and directly estimates the clean target frame using spatial and temporal information coming from both the target frame and its adjacent ones.

DeBlurNet is composed of three types of layers: down-convolutional layers compress the spatial resolution of the features while increasing the number of feature maps; flat-convolutional layers perform non-linear mapping while preserving both the spatial resolution and the number of feature maps; up-convolutional layers increase the spatial resolution while reducing the number of feature maps.

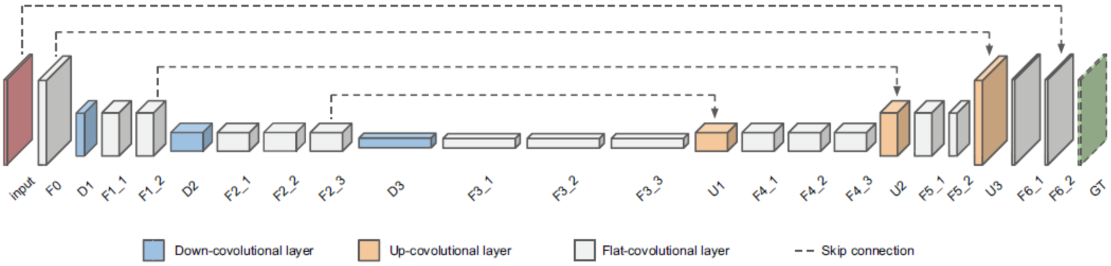


Figure 3.8: Architecture of DeBlurNet for video deblurring [17]

Su et al. [17], with their work, demonstrated that high-quality results could be obtained even without explicitly estimating pixel motion and performing frame alignment.

3.2.3 VESPCN

In 2018, Caballero et al. [18] combined the efficiency of sub-pixel convolutions [9] with the performance of spatial transformer networks [16] to obtain a fast and accurate framework for video super resolution. They extended ESPCN [9], originally proposed for single image super resolution, to the video domain, creating VESPCN.

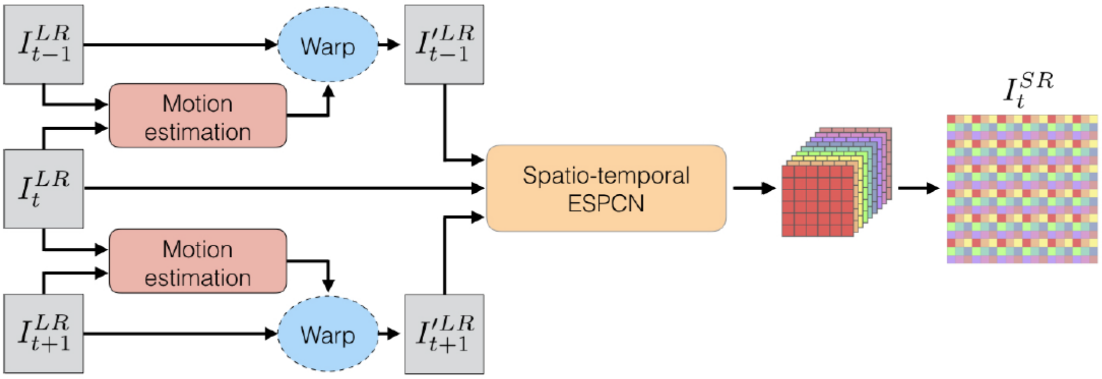


Figure 3.9: Architecture of VESPCN for video super resolution [18]

VESPCN performs pixel motion estimation using spatial transformer networks [16], uses bilinear interpolation to warp the neighboring frames to the target one, employs an adapted ESPCN [9], called spatio-temporal ESPCN, to fuse the information, performs upscaling using a sub-pixel convolutional layer [9] and produces the restored frame.

An overview of VESPCN is given by Figure 3.9. The pixel motion estimation module was designed following the coarse-to-fine approach, which is well known to be effective in estimating pixel motion in case of large motion.

VESPCN is end-to-end trainable because all its modules are differentiable. It is worthwhile to mention that Caballero et al. [18] studied how performance changes when more frames are used as input. They concluded that using more than five consecutive frames does not lead to any performance improvement.

3.2.4 DUF

Another contribution to video super resolution was given by Jo et al. [19] in 2018. Their network, DUF, implicitly uses motion information between consecutive frames to generate dynamic upscaling filters to upsample the target frame.

Since many approaches for video super resolution perform upscaling using bilinear or bicubic interpolation, which can hardly restore sharp details and textured regions, Jo et al. [19] introduced dynamic upscaling filters to perform the upscaling operation. Inspired by dynamic filter networks [20], dynamic upscaling filters are locally and dynamically generated based on the spatio-temporal neighborhood of each pixel in the low-resolution frames. Conceptually, dynamic upscaling filters are created based on pixel motions and, thus, they can produce better results than simply using bilinear or bicubic interpolation to perform upsampling. Each pixel in the low-resolution target frame has its own dynamic upscaling filters, whose number depends on the upscaling factor. Moreover, as the result of applying dynamic upscaling filters to a frame lacks of sharpness, DUF employs residual learning [6] to learn high frequency details that are then added to recover sharpness.

The proposed architecture is shown in Figure 3.10. As shown, DUF uses two different networks to generate dynamic upscaling filters and to learn high-frequencies. However, most of the weights between these two networks are shared.

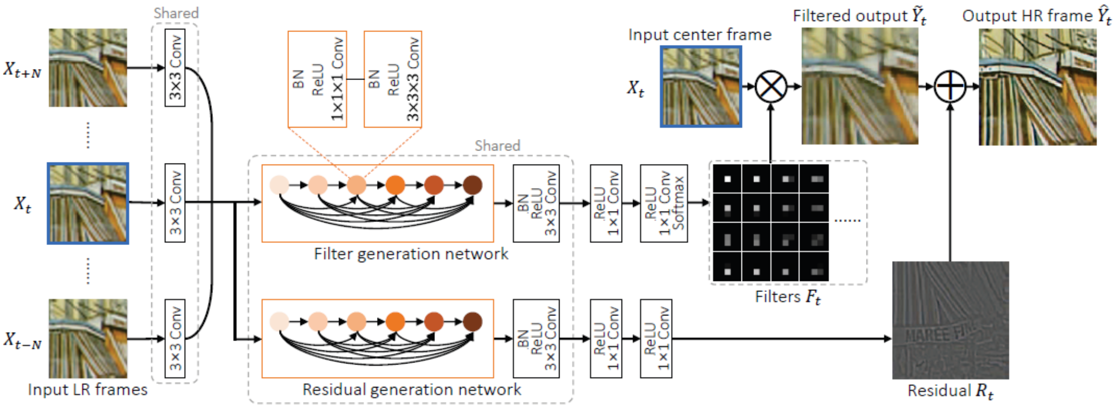
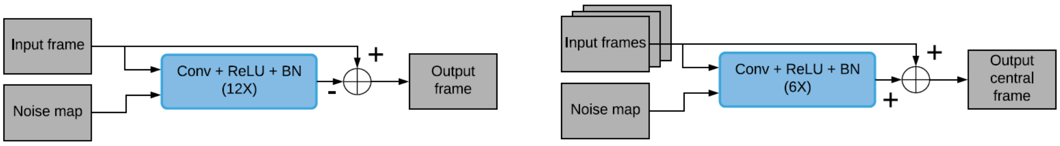


Figure 3.10: Architecture of DUF for video super resolution [19]

It is important to mention that, to capture spatio-temporal information, DUF uses 3D convolutions, as they are known to be more suitable than 2D convolutions to extract features from video data.

3.2.5 DVDNet

In 2019, Tassano et al. [21] introduced DVDNet for video denoising. Their framework consists of three steps: single image denoising, pixel motion estimation and warping and multiple image denoising. Single image denoising is performed using a spatial denoiser, whose structure is shown in Figure 3.11(a), considering one frame at a time. Such operation is important because, when dealing with noisy frames, pixel motion estimation becomes a very difficult task. So, removing some noise from each single frame is highly beneficial for the motion estimation step. In addition to the noisy frame, the spatial denoiser takes a noise map as input, encoding the estimated per-pixel standard deviation of the noise, which allows to better handle spatially varying noise. In order to estimate pixel motion between the target frame and a frame of its neighborhood and align them, DVDNet relies on DeepFlow [22]. Finally, when all the adjacent frames are warped to the target one, a temporal denoiser, shown in Figure 3.11(b), is used to capture spatio-temporal information and generate the restored frame. The noise map used for the spatial denoiser is also used for the temporal denoiser.



(a) Spatial denoiser (b) Temporal denoiser

Figure 3.11: Spatial and temporal denoisers used by DVDNet [21]

The overview of the DVDNet framework is reported in Figure 3.12. The spatial and temporal denoisers are trained separately: first, the spatial denoiser is trained to remove noise from each single frame, then, the temporal denoiser is trained using the neighboring frames warped to the target one to produce the final output.

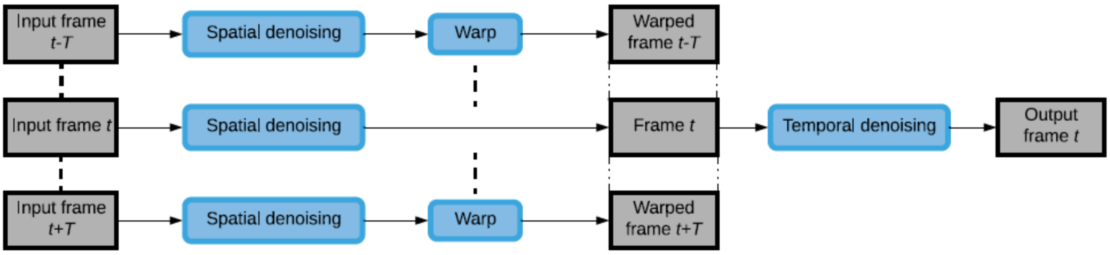


Figure 3.12: DVDNet framework for video denoising [21]

3.2.6 ViDeNN

Claus et al. [23], in 2019, extended DNCNN [5], originally proposed for single image denoising, to perform video denoising. Similarly to DVDNet [21], they designed a CNN, called ViDeNN, consisting of two denoisers: a spatial denoiser and a temporal denoiser. However, in contrast to DVDNet [21], ViDeNN does not use any specific module for pixel motion estimation and compensation, which are

implicitly handled by the network itself.
As illustrated in Figure 3.13, ViDeNN first performs single frame denoising, then the results are stacked and fed to the temporal denoiser producing the restored target frame. As in DVDNet [21], the spatial and temporal denoisers are trained separately.

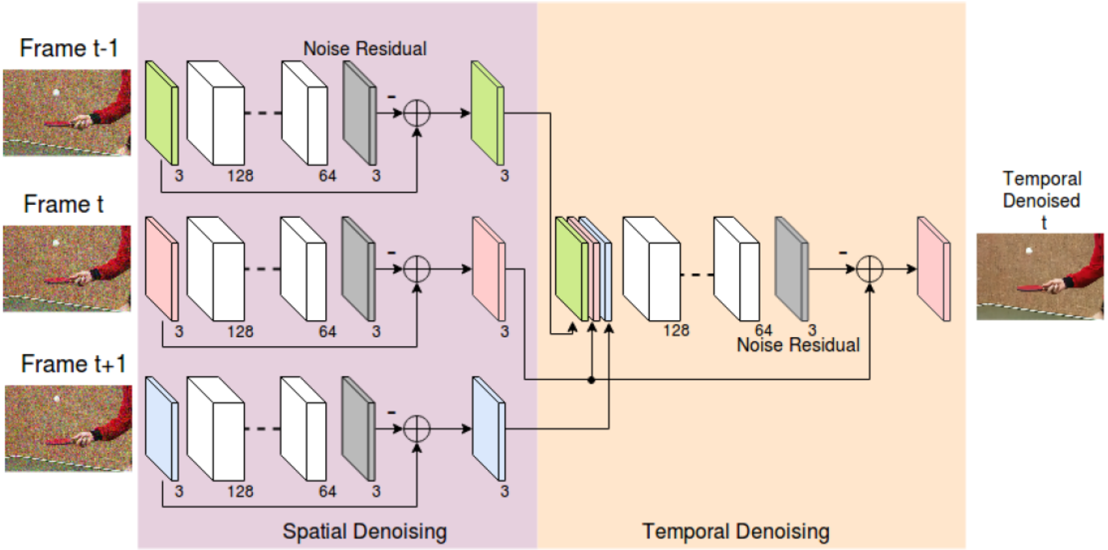


Figure 3.13: Architecture of ViDeNN for video denoising [23]

ViDeNN is able to perform blind Gaussian denoising, that is, it can remove Gaussian noise with unknown values of standard deviation.
In their work, Claus et al. [23] also studied how performance changes when more frames are used as input. As a result, they obtained that inputting more than three frames to ViDeNN does not lead to any performance improvement.

3.2.7 EDVR

In 2019, Wang et al. [24] won all the four tracks of the NTIRE19 video restoration and enhancement challenge [25], i.e. video super resolution, deblurring and compression artifact removal, with their EDVR.
The architecture of EDVR is illustrated in Figure 3.14. The cores of the proposed network are the alignment module, known as pyramid, cascading and deformable convolutions [26] (PCD), and the fusion module, known as temporal and spatial attention [27] (TSA).
The PCD module, shown in Figure 3.15(a), uses a coarse-to-fine approach to handle large motion. It is based on two well known principles in optical flow estimation: pyramidal processing and cascading refinement. In particular, at each level of the pyramid, deformable convolutions [26] are used to align a neighboring frame with the target one. Features of different frames are first aligned in lower scales and then, along with the learned offsets, are propagated to higher scales, enabling a

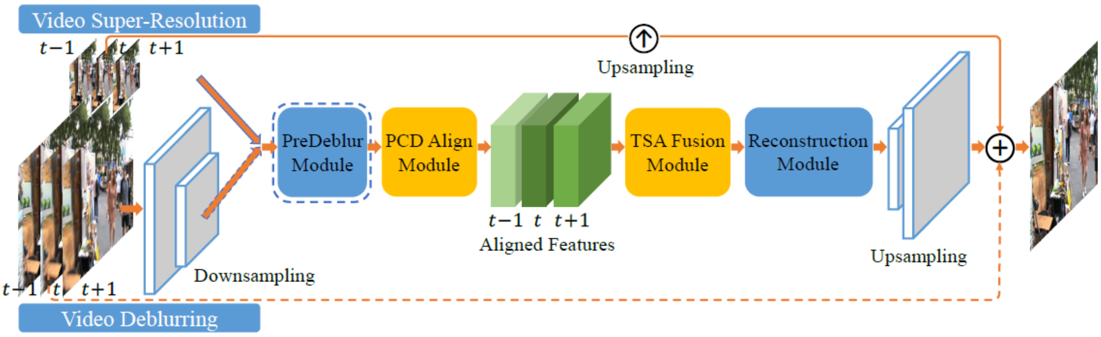
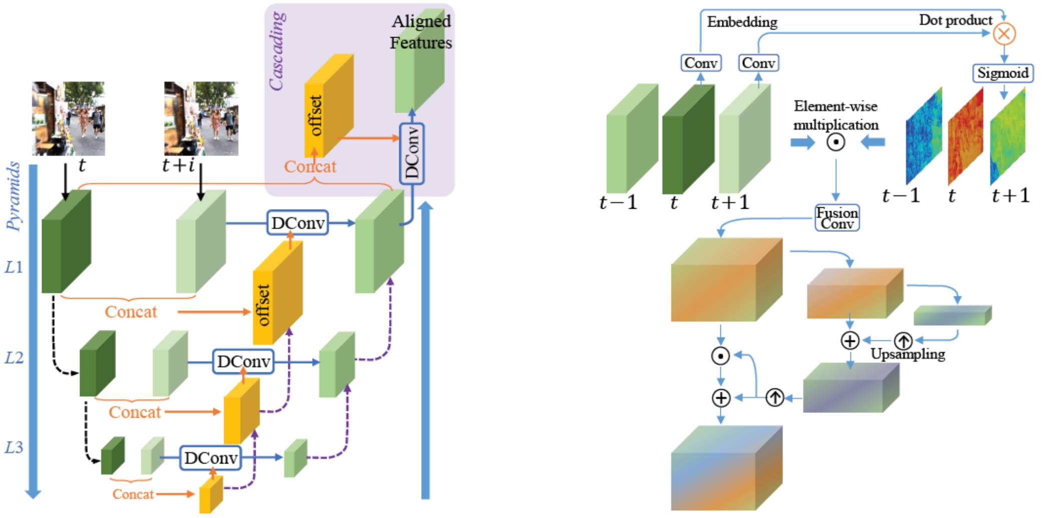


Figure 3.14: Architecture of EDVR for video super resolution, denoising, deblurring and compression artifact removal [24]

more precise motion compensation. An additional deformable convolution [26] is added to improve robustness of the alignment. It is worth mentioning that upscaling in the PCA module is performed using bilinear interpolation. In their experiments, they used three pyramid levels.



(a) Pyramid, cascading and deformable convolutions (PCD)

(b) Temporal and spatial attention (TSA)

Figure 3.15: Pyramid, cascading and deformable convolutions (PCD) and Temporal and spatial attention (TSA) modules used by EDVR [24]

The TSA module, shown in Figure 3.15(b), is used to aggregate information coming from the aligned frames. The idea behind temporal attention is that of weighting the features according to their importance, because not all the features coming from the aligned neighboring frames carry information useful to reconstruct the target frame. The same consideration applies also to spatial attention, because there are some spatial locations more useful than others and, hence, they must be weighed accordingly. More in detail, the temporal attention computes the element-wise

correlation between the features coming from the target frame and the features coming from one of its adjacent frames. The computed correlation is used to weight each feature of the neighboring frames, giving more weight to the features that are more similar to the ones of the target frame. Then, the obtained weighted features are fused and the spatial attention is computed, which assigns a weight to each location in each channel to better exploit cross-channel dependencies.
Besides the PCD and the TSA modules, EDVR uses a pre-deblur module to process each frame before aligning them, improving alignment accuracy. It also uses a final reconstruction module, which is a cascade of residual blocks that allows to perform the final refinement. For efficiency reasons, EDVR first downsamples the input frames to work in a low-resolution space. The spatial dimension is restored just at the end of the network.
Wang et al. [24] noticed that a single EDVR is able to obtain state-of-the-art performance in all the considered video restoration tasks. However, better performance can be obtained by applying a two-phase strategy: the outputs of the first EDVR are inputted to another EDVR. This strategy appears to be effective when dealing with severely distorted videos.
EDVR is end-to-end trainable and it is able to restore videos even in the presence of multiple artifacts.

3.2.8 FastDVDNet

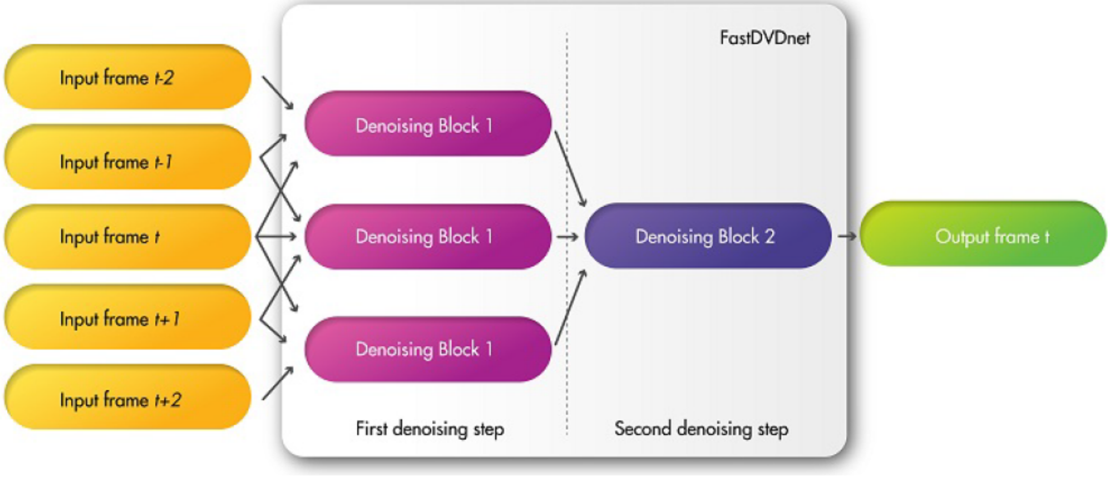


Figure 3.16: Architecture of FastDVDNet for video denoising [28]

More recently, in 2020, Tassano et al. [28] improved their DVDNet [21] proposing FastDVDNet. Instead of using an explicit motion estimation stage, FastDVDNet mainly improves DVDNet [21] by performing implicit motion estimation and compensation between frames. This allows FastDVDNet to remove artifacts caused by wrong motion estimations, also increasing its efficiency. Moreover, in contrast to their previous work in which the spatial and temporal denoisers are trained

separately, FastDVDNet is end-to-end trainable. Similarly to DVDNet [21], FastDVDNet takes the noise map of the target frame as input to better handle spatially varying noise.
The network proposed by the authors is a two-step restoration network: the first denoising step is composed of three denoising blocks taking a sequence of three frames and the noise map of the target frame as input, while the second denoising step consists of a single denoising block taking the outputs of the denoising blocks of the first denoising step and the same noise map as input. Note that all the three denoising blocks of the first denoising step share the same weights.
The architecture of FastDVDNet is shown in Figure 3.16, while Figure 3.17 shows the detailed composition of each denoising block. As shown, the upscaling process is done by means of sub-pixel convolutional layers [9]. FastDVDNet takes five frames as input and, as shown in Figure 3.16, each denoising block of the first denoising step always uses information about the target noisy frame, as it is inputted to each denoising block.

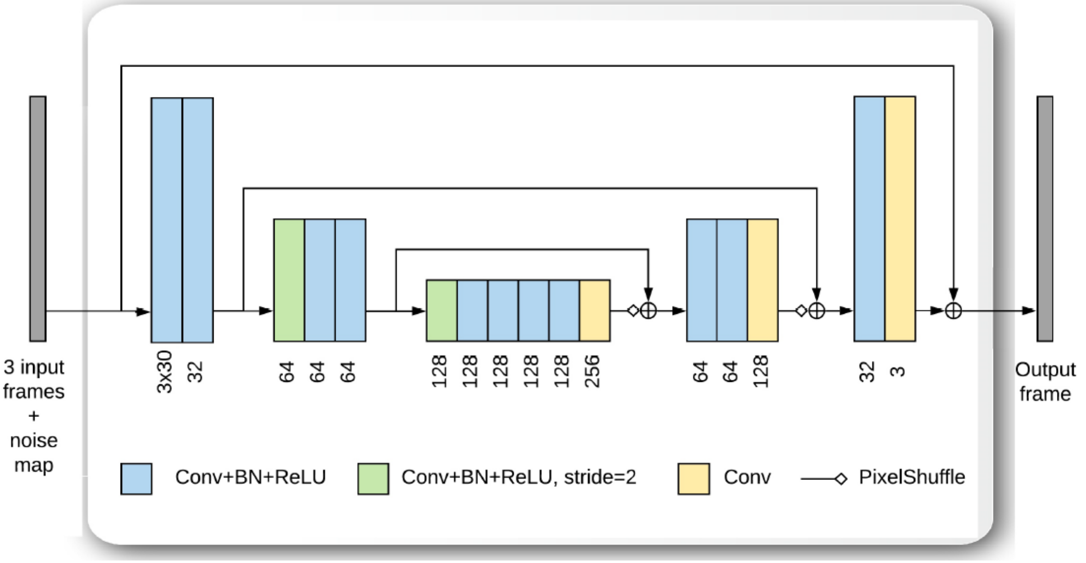


Figure 3.17: Denoising block used by FastDVDNet [28]

In their work, Tassano et al. [28] also studied the impact that applying a single-step restoration rather than a two-step restoration has on the denoising performance, concluding that the two-step restoration allows to obtain better results.

3.2.9 STDF

Deng et al. [29], in 2020, proposed a new framework to remove compression artifacts from videos. They introduced STDF, a new spatio-temporal deformable fusion schema to remove compression artifacts based on the idea of deforming the spatio-temporal sampling positions of standard convolutions, making them able to capture more relevant information. Besides, they avoided explicit motion estimation so that the restoration process is not affected by wrong estimations. In contrast to previous approaches, which consider just a limited neighborhood of a given target frame, STDF considers the entire video sequence.

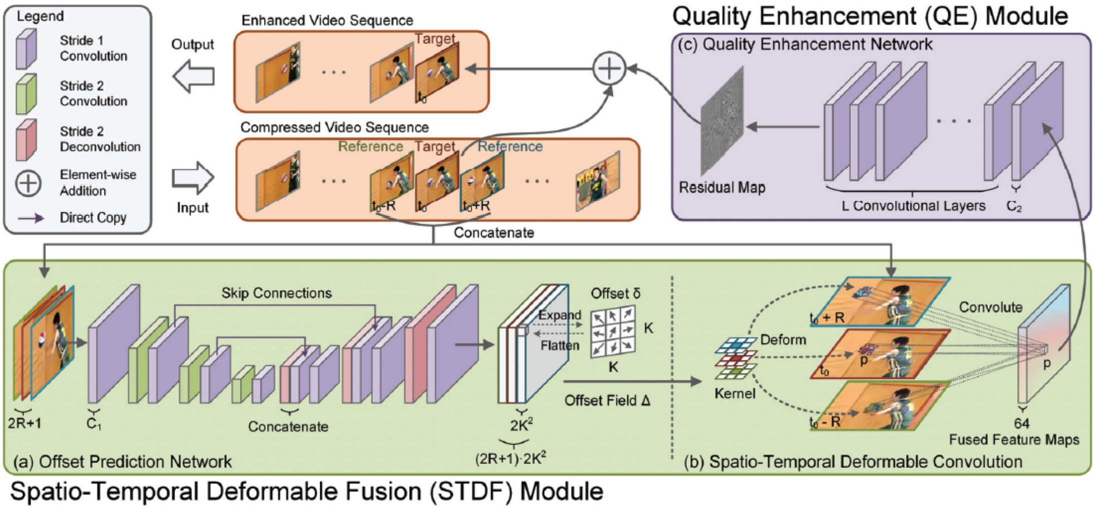


Figure 3.18: STDF framework to remove compression artifacts from videos [29]

As illustrated in Figure 3.18, STDF is composed of two modules: the spatio-temporal deformable fusion (STDF) module and the quality enhancement (QE) module. The first module uses deformable convolutions [26] to learn position-specific offsets that allow to deform standard convolution, making them able to model complex geometric transformations. This allows STDF to pay more attention to motion cues. The second module is a simple CNN that performs the restoration of the target frame using the information extracted by the previous module.
As done by previous works, the network uses residual learning [6] to predict the residual that is added to the compressed target frame to obtain the final result.

3.2.10 MFQE 2.0

Guan et al. [30], in 2020, improved their previous work [31] obtaining higher performance on the task of video compression artifact removal. Their work is based on the idea that, when dealing with compression artifacts arising from inter-frame compression methods, there are some frames that are more compressed than others. For this reason, given the target frame, it could be useful to find its high-quality adjacent frames because they may contain information useful to reconstruct it. Such high-quality neighboring frames are called “peak quality frames” (PQFs). Following this idea, they designed a framework that first searches for the previous and the subsequent PQFs of the target frame, then aligns them and finally uses the information contained in the aligned PQFs to restore the target frame.
The overview of their framework, named MFQE 2.0, is illustrated in Figure 3.19. MFQE 2.0 is composed of three modules: a biLSTM based PQF detector (PQF

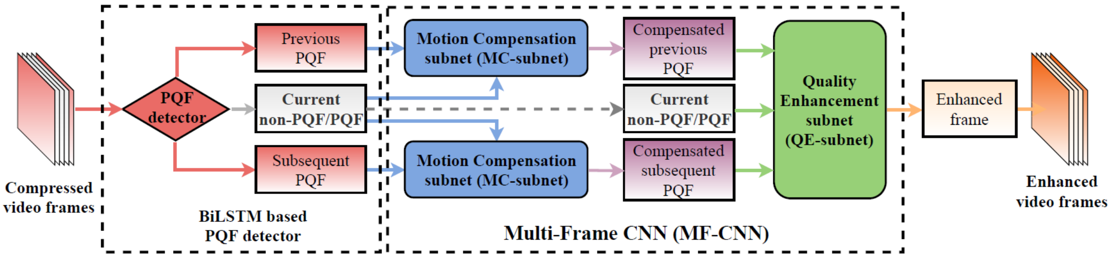
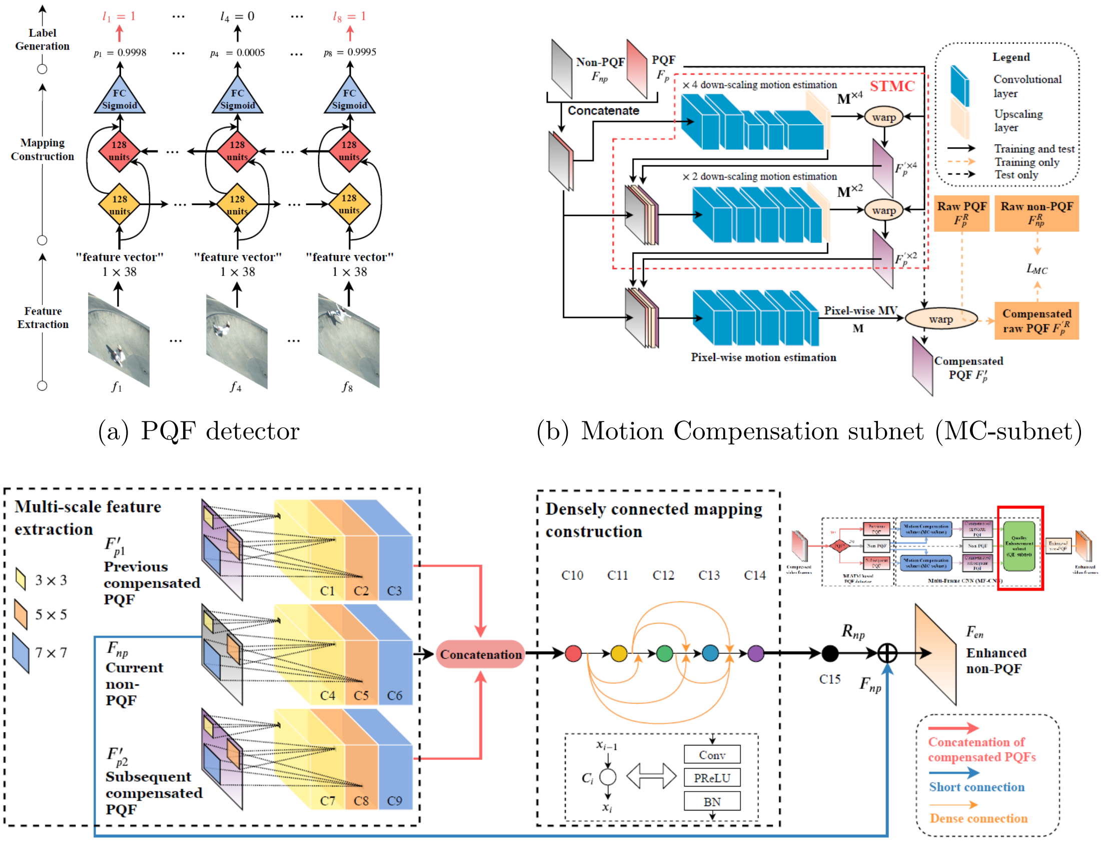


Figure 3.19: MFQE 2.0 framework to remove compression artifacts from videos [30]



(a) PQF detector

(b) Motion Compensation subnet (MC-subnet)

(c) Quality Enhancement subnet (QE-subnet)

Figure 3.20: PQF detector, Motion Compensation subnet (MC-subnet) and Quality Enhancement subnet (QE-subnet) used by MFQE 2.0 [30]

detector), a motion compensation subnet (MC-subnet) and a quality enhancement subnet (QE-subnet). The first module, shown in Figure 3.20(a) is a bidirectional LSTM [32] used to extract and model frame correlation. Thanks to its bidirectionality, information can be extracted in both directions. Given the target frame, the purpose of this module is that of computing the probability of its neighboring frames of being PQFs. Such probabilities are then refined to be sure there is at least a previous and a subsequent PQF of a given frame in a specified temporal window. It is worth mentioning that frames are not directly fed to this module, but 38 features related to compression domain and quality assessments methods [33] are extracted and used to detect PQFs. The MC-subnet module, shown in Figure 3.20(b) performs motion estimation and compensation between the target frame and the PQFs found by the previous module. This is done by means of a modified spatio-temporal motion compensation architecture proposed by Caballero et al. [9], in which Guan et al. [30] added the computation in full resolution to improve alignment accuracy. Finally, when the PQFs are warped to the target frame, the QE-subnet module, shown in Figure 3.20(c), performs multi-scale feature extraction and restores the target frame.

3.2.11 EVRNet

Very recently, at the end of 2020, EVRNet was proposed by Mehta et al. [34]. EVRNet, which literally means “efficient video restoration network”, is able to perform different video restoration tasks using few computational resources and in a very fast way, making it suitable to run on edge devices. It is capable of dealing with video super resolution, video denoising and compression artifact removal.
Figure 3.21 shows the EVRNet architecture. The proposed architecture consists of

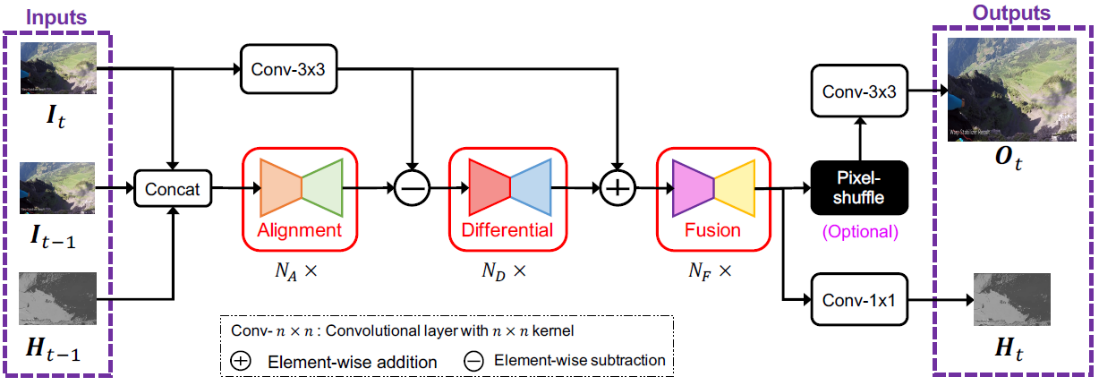
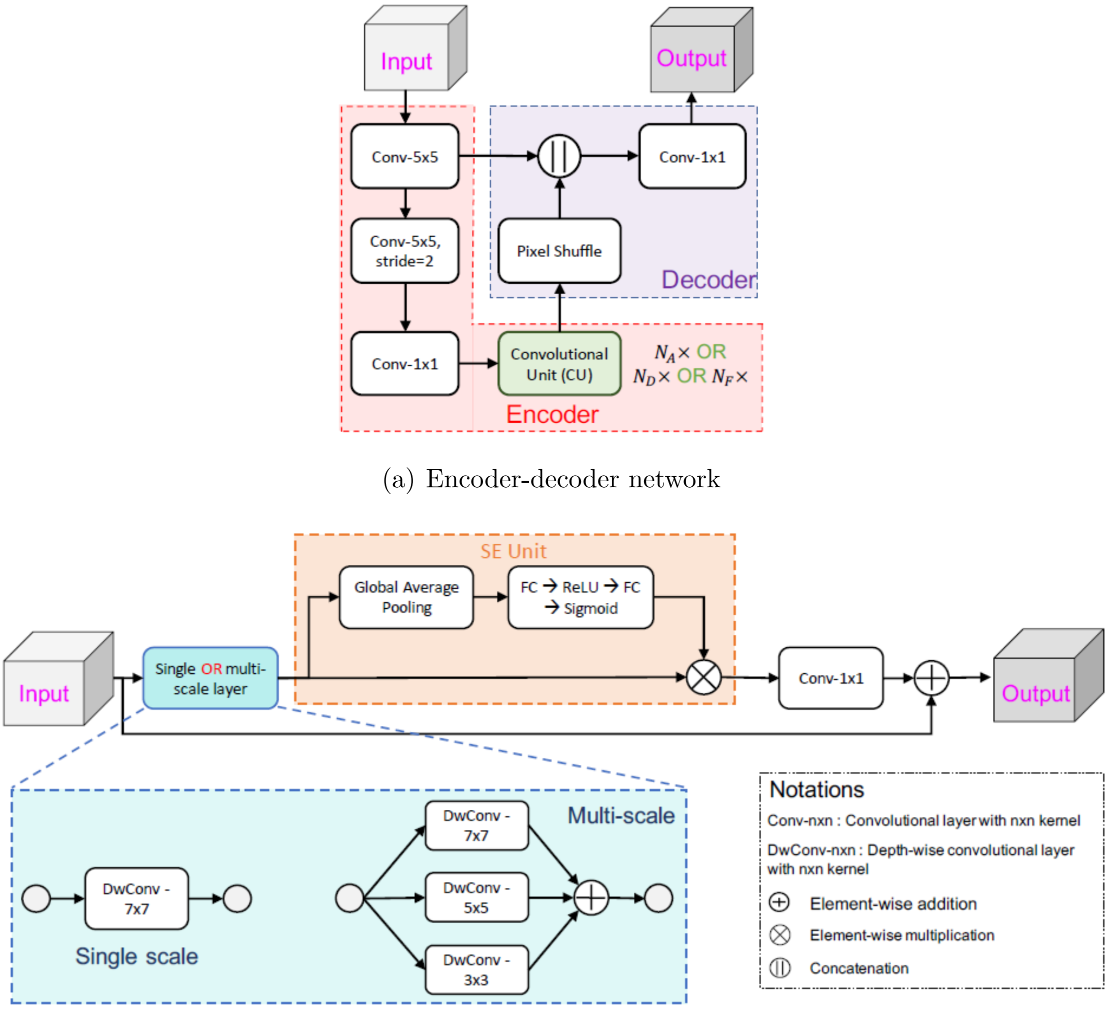


Figure 3.21: Architecture of EVRNet for video super resolution, denoising and compression artifact removal [34]

three modules: alignment, differential and fusion. The alignment module is used to implicitly align frames using a pyramidal structure, allowing EVRNet to handle large motion without the need to compute optical flow. The differential module aims to learn high-frequency components, such as edges. Finally, the fusion module combines the high-frequency components learned by the differential module and the input target frame projected in an embedding dimension to produce the final result.
Each module is a light-weight and shallow encoder-decoder architecture, as shown in Figure 3.22(a). They are equal in construction, what changes is the number of convolutional units (CU), shown in Figure 3.22(b).
The differential and fusion modules are inspired by traditional image enhancement



(a) Encoder-decoder network

(b) Convolutional unit (CU)

Figure 3.22: Encoder-decoder network and convolutional unit (CU) used by EVRNet [34]

methods, such as unsharp masking [35].
As it is possible to see from Figure 3.21, EVRNet outputs two frames: the restored target frame and the latent target frame, which corresponds to the restored target frame in an embedding dimension. Such latent target frame is used to predict the next frame, making EVRNet auto-regressive, i.e. the output at time t is used as input at time $t+1$.

3.3 Contrast enhancement

Contrast enhancement is an image processing technique that aims to improve the perceptibility of objects in the scene by enhancing the brightness difference between objects and their backgrounds.

In the literature, several methods have been proposed to enhance contrast in both images and videos. Such methods can be classified into three main categories: histogram equalization methods try to enhance contrast using image histograms; Retinex theory [36] based methods decompose the image into reflectance and illumination, and contrast enhancement consists in manipulating the estimated illumination; deep learning based methods enhance contrast by means of deep neural networks, which learn the optimal operations given pairs of input and output images.
Since 2017, researchers all over the world have started investing many resources to build framework for contrast enhancement using deep learning techniques. The following approaches have been introduced to address the problem of low-light image and video enhancement, a branch of contrast enhancement that deals with images and videos captured in low-light conditions.

3.3.1 LLCNN

Tao et al. [37] were the first to propose to use deep learning to enhance low-light images, in 2017. Their network, LLCNN, is able to learn how to filter low-light images with different kernels and combine multi-scale feature maps to generate images that seem to be captured in normal-light conditions.
Figure 3.23 shows the architecture of LLCNN. The first convolutional layer is used to preprocess the input image and produce uniform input, while the last layer generates the enhanced image. Between these two layers, several convolutional modules are placed. Such convolutional modules, shown in Figure 3.24, are inspired by Inception [38] and ResNet [6] and are designed to cope with the vanishing gradient problem.

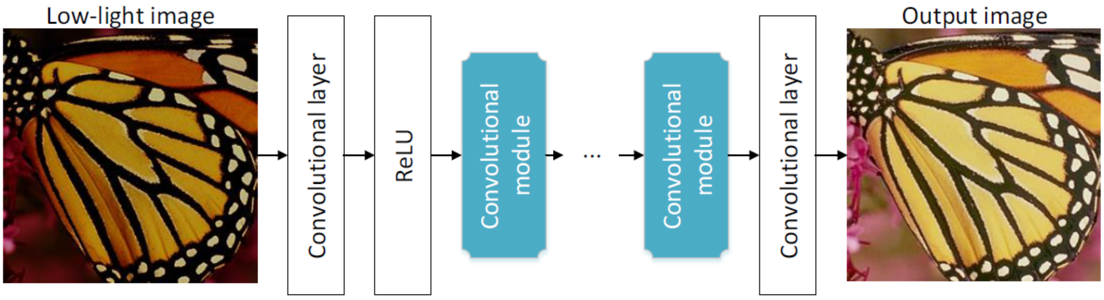


Figure 3.23: Architecture of LLCNN for low-light image enhancement [37]

Since preserving textures in low-light image enhancement is very important and the brightness is allowed to fluctuate around the ground truth, Tao et al. [37] stated that SSIM [12] is the most suitable metric to optimize. For this reason, they adopted a loss based on SSIM [12].
In their experiments, the authors tried to train their network with different number of convolutional modules, concluding that the deepest LLCNN was the most effective one in enhancing low-light images.

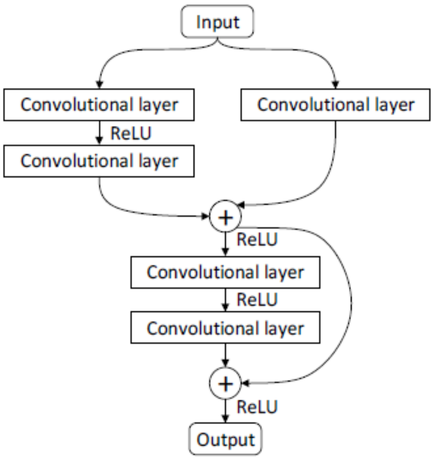


Figure 3.24: Convolutional module used by LLCNN [37]

3.3.2 MBLLEN

Lv et al. [39], in 2018, addressed the main limitation of LLCNN [37], that is, it did not consider the fact that noise may be introduced by camera sensors when images are captured in low-light conditions. To do this, they proposed MBLLEN, which is able to enhance low-light images while removing the noise introduced by poor acquisition conditions.

MBLLEN is shown in Figure 3.25. It decomposes the enhancement problem into different sub-problems related to different feature levels, which are solved to produce the enhanced image via multi-branch fusion.

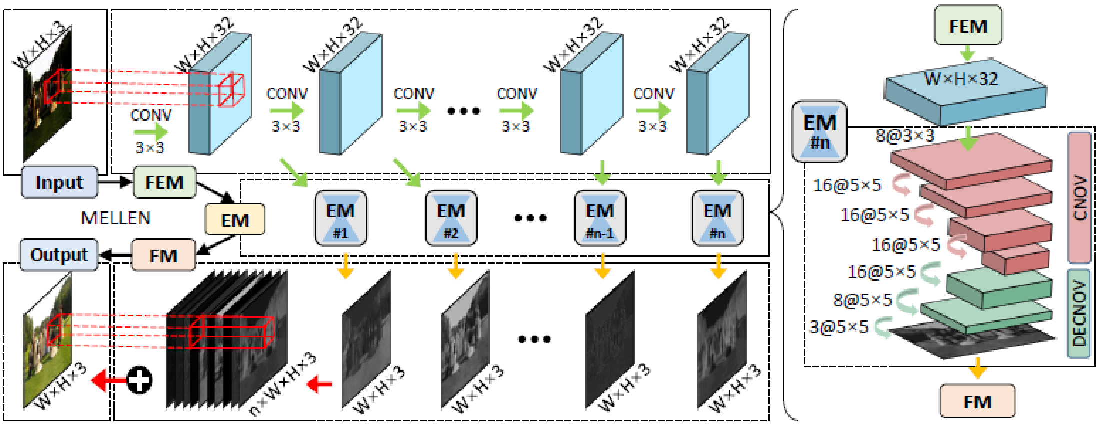


Figure 3.25: Architecture of MBLLEN for low-light image enhancement [39]

The proposed network is composed of three modules: the feature extraction module (FEM), the enhancement module (EM) and the fusion module (FM). The first module uses several convolutional layers and non-linear mapping to extract feature maps for the other modules. The second module contains multiple subnets whose number equals the number of layers in FEM. Each subnet is an encoder-decoder network that uses convolutions and upconvolutions to enhance the input image at different feature levels. Note that all these subnets have the same structure and are trained together with the rest of the architecture, but they are actually different networks, that is, they do not share any weight. Finally, the last module fuses the outputs of all the subnets in EM to merge the information and produce the enhanced image.

In addition to MBLLEN, Lv et al. [39] proposed a novel loss function considering structure information, context information and regional difference in the image: the structural loss is based on SSIM [12]; the context loss is based on high-level information that can be extracted using another CNN; the region loss is used to pay more attention to low-light regions.

MBLLEN, as stated by its authors, can be easily extended to videos by replacing 2D convolutions with 3D convolutions in the feature extraction and enhancement modules.

3.3.3 GLADNet

In 2018, Wang et al. [40] proposed GLADNet, a deep neural network that exploits global illumination information and detail preservation to enhance low-light images.

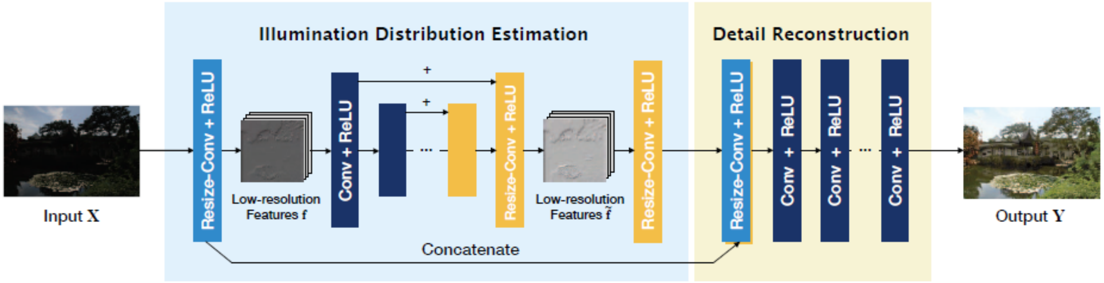


Figure 3.26: Architecture of GLADNet for low-light image enhancement [40]

GLADNet can be divided into two modules, as shown in Figure 3.26: a global illumination prediction module and a detail reconstruction module. The global illumination prediction module applies an encoder-decoder network that reduces the spatial dimension in such a way that the receptive field at the bottleneck will be large enough to cover the entire image. This allows GLADNet to acquire a global awareness of the whole illumination distribution. Then, the feature maps are upscaled to obtain the feature maps for illumination prediction. To improve efficiency, the input image is downscaled before feeding it to the network. The feature maps obtained from the first module are used by the detail reconstruction module, along with the input image, to reconstruct the details lost during the downscaling process. This module outputs the enhanced image.

3.3.4 CNN with simple reflection model

In 2019, Moon et al. [41] proposed a locally adaptive contrast enhancement method using a CNN and a simple reflection model.

The proposed framework consists of four steps, as shown in Figure 3.27. The first step is a low-contrast estimation step that generates the low-contrast probability map to identify low-contrast regions. To do this, they created an encoder-decoder network that takes the low-light image as input and produces the low-contrast probability map as output. The second step is to create a contrast gray scale map from the input image. This is achieved by using saliency-guided decolorization methods [42] that are able to properly express contrast information. Then, the third step is that of refining the low-contrast probability map to prevent halos and undesired over-enhancement artifacts. This is done by means of guided filter [43], which uses the contrast gray scale map as guide image. Finally, the last step is to enhance the input image using the information contained in the refined low-contrast probability map, following the reflection model.

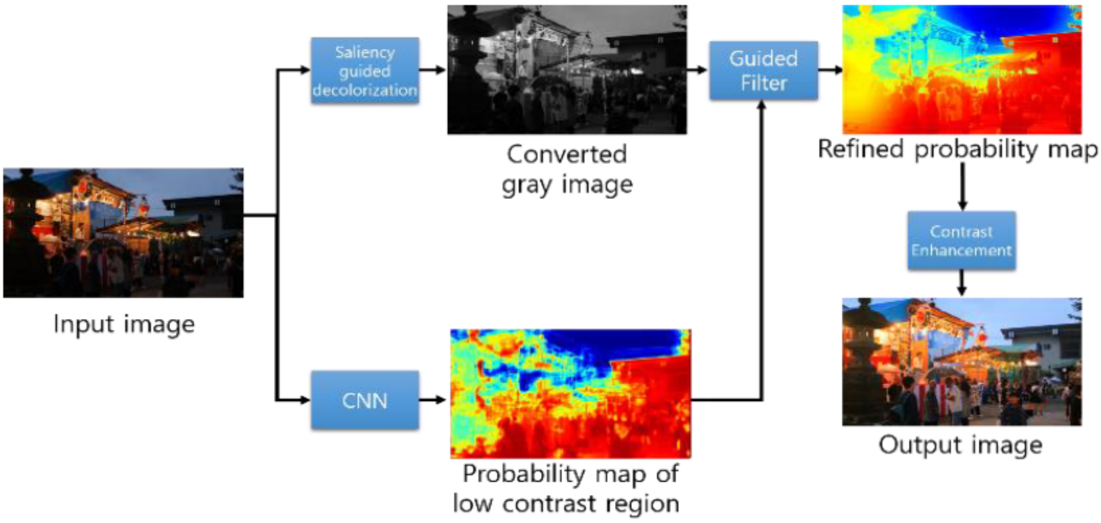


Figure 3.27: Image enhancement framework proposed by Moon et al. for low-light image enhancement [41]

3.3.5 Retinex-GAN

Shi et al. [44], in 2019, presented a framework to enhance low-light images based on Retinex theory [36] and Generative Adversarial Networks [11] (GANs). In Retinex theory [36], an image S is given by $I \times R$, where I is the illumination image and R is the reflectance image. Hence, the objective here is to decompose the input image S into I and R and enhance I to obtain the final result.

The proposed framework, as illustrated in Figure 3.28, is composed of a generator and a discriminator. The generator includes the decomposition step, which aims to split the input image into the illumination and the reflectance components, and the

enhancement step, which tries to enhance the brightness of the image. Then, the reflectance image and the enhanced illumination image are recombined to obtain the final enhanced image. The discriminator is used to make the generated image look more realistic.
To improve the results, the authors designed a multi-task loss that combines adversarial loss, SSIM [12] loss and L1 loss.

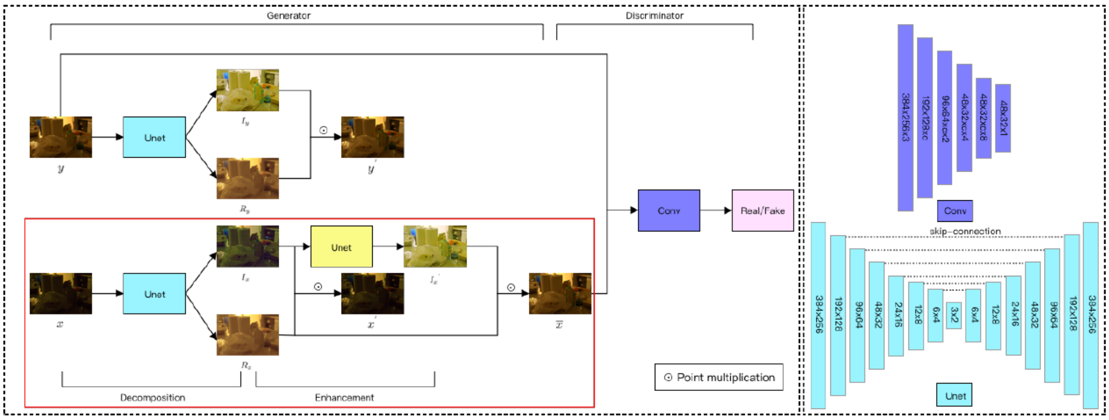


Figure 3.28: Retinex-GAN framework for low-light image enhancement [44]

3.3.6 ALEN

Zhang et al. [45], in 2020, presented a novel attention-based neural network to produce high-quality enhanced images from raw sensor data.

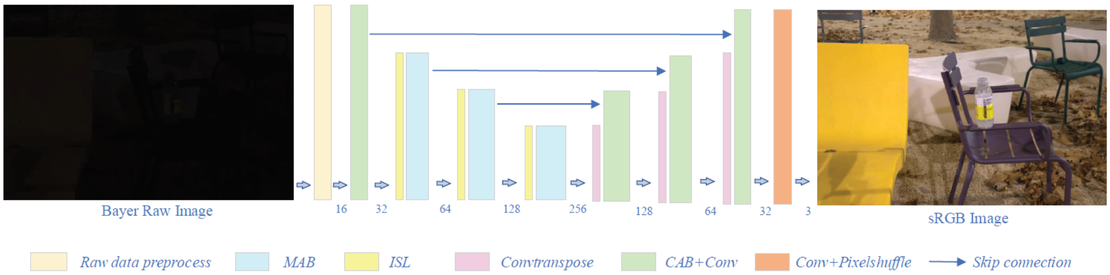
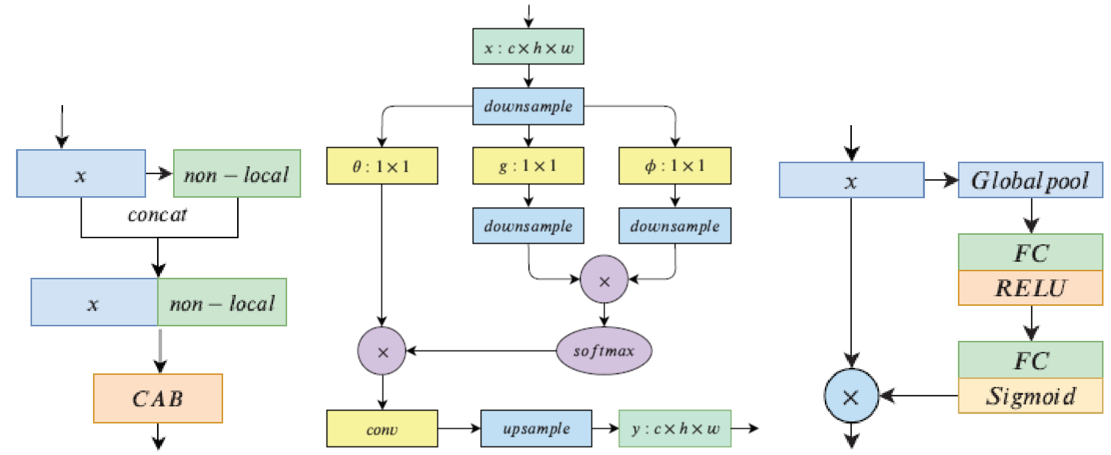


Figure 3.29: Architecture of ALEN for low-light image enhancement [45]

The proposed network, named ALEN and shown in Figure 3.29, uses spatial and channel attention to take into account both local and global information. Spatial attention is performed using non-local operations [46], which are used to make the network have a global receptive field by aggregating different position information in a feature map. This because the authors observed that a large receptive field is fundamental to reduce color artifacts, but increasing it using several convolutional layers is inefficient. The non-local operation is shown in Figure 3.30(b). Channel attention, performed using Squeeze-Excitation blocks [47], weights the contribution of each feature map, allowing ALEN to refine redundant color features. The block used for channel attention is shown in Figure 3.30(c). Spatial and channel attention are applied in ALEN using a mixed-attention, as shown in Figure 3.30(a), which first uses spatial attention to obtain features with a wider range of information in the spatial domain and then uses channel attention to generate the final feature representation.



(a) Mixed-attention block (b) Non-local operation (c) Channel-attention block

Figure 3.30: Mixed-attention block, non-local operation and channel-attention block used by ALEN [45]

Moreover, to reduce the information loss and select useful features, they replaced max pooling layers with a novel inverted shuffle layer (ISL), which performs the inverted operation of the Pixel Shuffle layer [9].
As other approaches, the final loss combines both the L1 loss and the SSIM [12] loss.

3.3.7 DALE

In 2020, Kwon et al. [48] introduced DALE, a new dark region-aware low-light image enhancement framework that recognizes dark regions in the input image and intensively enhances their brightness.
DALE, as shown in Figure 3.31, consists of two networks: a visual attention network and an enhancement network. The visual attention network produces an attention map able to recognize dark regions. To train such network, the authors created a dataset using a local illumination synthesis method based on super-pixels, which are randomly darkened. Then, the enhancement network takes the low-light image and the estimated visual attention map of that image as input, and produces the enhanced image.
It is worth mentioning that these two networks are trained separately.

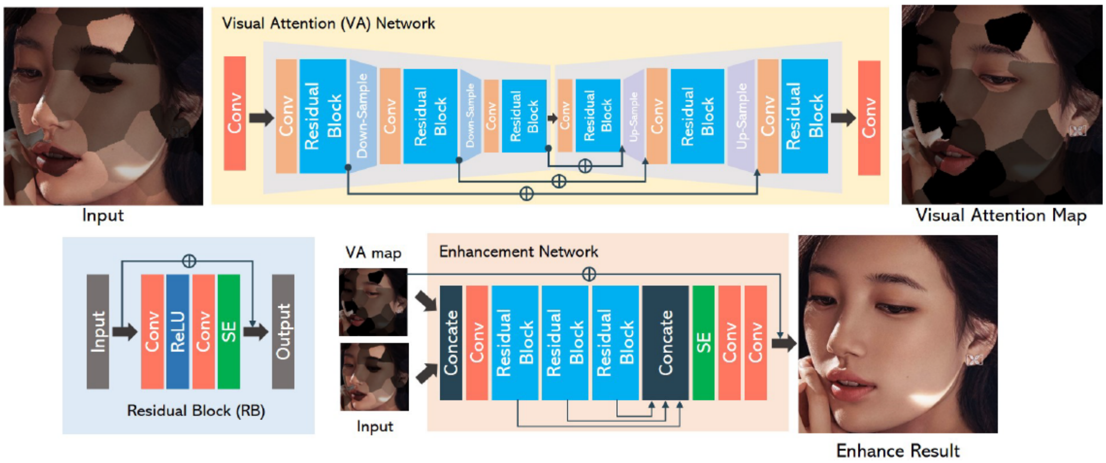


Figure 3.31: DALE framework for low-light image enhancement [48]

3.3.8 NEID

At the beginning of 2021, Jiang et al. [49] introduced NEID, a deep neural network that enhances both brightness and details of low-light images simultaneously. NEID is able to perform both low-light image enhancement and super resolution, producing normal-light images with rich details and high visual quality.

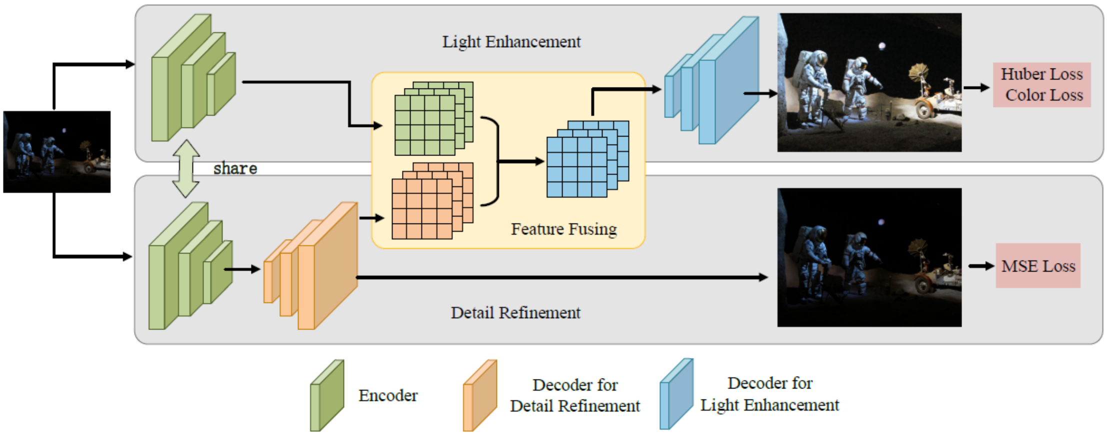


Figure 3.32: NEID architecture for simultaneous low-light image enhancement and image super resolution [49]

In brief, as it is possible to see in Figure 3.32, NEID is a two-stream network consisting of two branches deployed in parallel: the Light Enhancement branch (LE) and the Detail Refinement branch (DR). The LE and DR branches extract features that are then fused by a Feature Fusion module (FF). The Light Enhancement branch, as the name suggests, is used to enhance the brightness of the input image. However, since the LE branch is not able to recover high-resolution and detail-rich images from the low-resolution input, as the upscaling module used in the LE branch does not bring enough information, the Detail Refinement branch is used

to reconstruct fine-grained detail information. To enhance the brightness and produce a high-resolution version of the input image, the outputs of the LE and DR branches are fused by means of the FF module, which weights the features obtained by the LE branch guided by the features obtained by the DR branch.
To train their network, Jiang et al. [49] combined the Huber loss [50], the MSE loss and the Color loss.

3.3.9 Zero-DCE

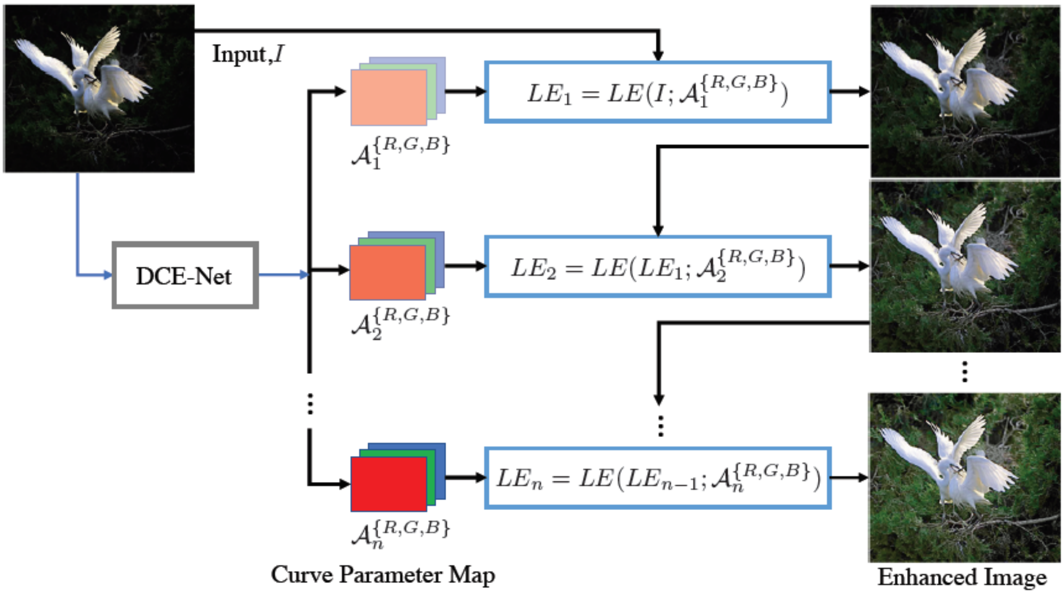

Input,I
DCE-Net
$LE_1 = LE(I; \mathcal{A}_1^{\{R,G,B\}})$
$\mathcal{A}_1^{\{R,G,B\}}$
$LE_2 = LE(LE_1; \mathcal{A}_2^{\{R,G,B\}})$
$\mathcal{A}_2^{\{R,G,B\}}$
$LE_n = LE(LE_{n-1}; \mathcal{A}_n^{\{R,G,B\}})$
$\mathcal{A}_n^{\{R,G,B\}}$
Curve Parameter Map
Enhanced Image

Figure 3.33: Zero-DCE framework for low-light image enhancement [51]

Very recently, in 2021, Li et al. [51] presented a novel method to enhance low-light images formulating the problem of light estimation as a task of image-specific curve estimation using deep learning. One of the main advantages of the proposed framework, called Zero-DCE and shown in Figure 3.33, is that of being zero-reference, i.e. it does not require any paired or unpaired data for the training process. This is possible thanks to a carefully designed non-reference loss function, which takes into account a spatial consistency loss, an exposure control loss, a color constancy loss and an illumination smoothness loss.
In brief, Zero-DCE uses a deep neural network to estimate pixel-wise and high-order curves for dynamic range adjustment of a given low-light image. Then, the framework iteratively maps all the pixels of the input image to new pixels according to the estimated curves to obtain the enhanced image.
The key components of Zero-DCE are the light-enhancement curve, which is monotone and differentiable, the deep neural network DCE-Net, which is used to estimate the pixel-wise curve parameter maps to be applied to the input image to obtain its enhanced version, and the non-reference loss function, which enables non-reference learning in DCE-Net.

Chapter 4

Methodological approach for video restoration

This chapter presents the methodological approach followed in this thesis in order to build a new deep video restoration network able to restore videos even in the presence of multiple artifacts.
It is organized as follows. Section 4.1 describes the main characteristics and components of video restoration approaches that emerged from a deep analysis of the approaches presented in Chapter 3. Section 4.2 presents the motivations that supported the decision of using FastDVDNet [28] as baseline model, whereas Section 4.3 explains this architecture more in detail, better clarifying its main components. Finally, Section 4.4 details the approach proposed in this thesis work, named Multi-distorted Video Restoration Network, to restore videos affected by multiple distortions.

4.1 Characteristics and components for video restoration

Several state-of-the-art approaches for video restoration have been presented in Chapter 3. An in-depth analysis of these methods is needed in order to have a clear picture about how some important operations are performed and to understand the common ideas behind every video restoration framework. Such analysis concerns how the methods under study perform motion estimation and frame alignment, how they address the problem of different artifact intensities, what their basic components are, such as architectural structures and the way they perform specific operations, and, finally, the loss functions they use.
In general, a video restoration framework commonly performs the following operations:

1. single frame restoration (optional): each video frame is restored independently in order to make the next steps easier;

2. motion estimation: given the target frame and its adjacent frames, the pixel motion between them is estimated, that is, each pixel in the target frame is detected within its neighboring frames in order to compute a motion vector representing how pixels have moved;

3. frame alignment: using the pixel motion estimation computed by the previous step, each adjacent frame is warped to the target one, i.e. it is transformed into the viewpoint of the target frame;

4. information extraction and fusion: given the target frame and its aligned adjacent frames, spatial and temporal information is extracted and fused to restore the target frame.

The first operation is marked as optional because it is not performed by all the methods, but some of them initially restore single frames because, as they report, the motion estimation operation is known to be challenging, and it could become even more difficult when frames are severely distorted. Examples of methods that try to restore single frames as first step are DVDNet [21] and ViDeNN [23], as already mentioned in Subsection 3.2.5 and 3.2.6.

4.1.1 Motion estimation and frame alignment

The first analysis regards how the state-of-the-art approaches estimate pixel motion and use such estimation for frame alignment. This allows to understand whether there is a specific technique for these operations that is more suitable for a particular restoration task.

Motion estimation and frame alignment are fundamental tasks that any video restoration approach must address. Motion estimation is the process of determining motion vectors describing the transformation from a 2D image to another, usually from adjacent frames in a video sequence. Frame alignment consists in using the estimated motion vectors to warp the source frame to the target one, transforming the former into the viewpoint of the latter.

All the video restoration approaches can be divided into two classes based on how they perform these operations, which can be explicitly or implicitly done by the network. In this section, for simplicity reasons, the methods performing implicit motion estimation and alignment will be simply called “implicit” approaches, while the others “explicit” approaches.

On the one hand, explicit methods have a specific step in their framework whose purpose is to compute pixel motion and use these motion vectors to perform frame alignment, allowing to extract spatially precise information from multiple frames. However, when the estimated pixel motion is wrong, they may introduce visible artifacts. On the other hand, implicit methods do not have any explicit step to estimate pixel motion and perform frame alignment, which are implicitly computed by the network during the restoration process. Such methods do not suffer from the aforementioned problem, but they may produce worse results than explicit methods when the estimated pixel motion is correct.

Table 4.1 reports the methods used by the analyzed approaches for pixel motion estimation and frame alignment.

Table 4.1: Methods used by the state-of-the-art approaches to estimate pixel motion and perform frame alignment

Approach name	**Task(s)**	**Implicit/Explicit**	**Motion estimation**	**Frame alignment**
TOFlow [14]	Super resolution, Denoising, Frame interpolation, Compression artifact removal	Explicit	SpyNet [15]	Spatial transformer network [16]
DeBlurNet [17]	Deblurring	Implicit	Encoder-decoder network	
VESPCN [18]	Super resolution	Explicit	Spatial transformer network [16]	
DUF [19]	Super resolution	Implicit	3D convolutions	
DVDNet [21]	Denoising	Explicit	DeepFlow [22]	
ViDeNN [23]	Denoising	Implicit	Residual network [6]	
EDVR [24]	Super resolution, Denoising, Deblurring, Compression artifact removal	Implicit	Deformable convolutions [26]	
FastDVDNet [28]	Denoising	Implicit	Encoder-decoder network	
STDF [29]	Compression artifact removal	Implicit	Deformable convolutions [26]	
MFQE 2.0 [30]	Compression artifact removal	Explicit	Spatial transformer network [16]	
EVRNet [34]	Super resolution, Denoising, Compression artifact removal	Implicit	Encoder-decoder network	

The vast majority of the analyzed approaches implicitly estimate pixel motion and perform frame alignment. Regarding the explicit approaches, TOFlow [14] and DVDNet [21] use an external network specifically designed for this task. More in detail, TOFlow [14] uses SpyNet [15] to perform motion estimation whereas DVDNet [21] relies on DeepFlow [22] both for motion estimation and frame alignment. However, these networks are not designed to estimate pixel motion in the case of degraded videos, thus they may fail the estimation process when the artifacts are severe. Other approaches, that is, MFQE 2.0 [30] and VESPCN [18] use spatial transformer networks [16], which have shown good performance for this task and, importantly, they can be end-to-end trained with the entire framework. Concerning the implicit approaches, many different methods are used. DeBlurNet [17], FastDVDNet [28] and EVRNet [34] perform motion estimation and frame alignment by means of encoder-decoder networks, EDVR [24] and STDF [29] rely on deformable convolutions [26], DUF [19] exploits 3D convolutions and ViDeNN uses a residual network.

It is possible to conclude that there is no relationship between the motion estimation and compensation methods used by the approaches and the considered restoration task. In addition, most of the approaches adopt an implicit technique.

4.1.2 Blind methods and non-blind methods

The second analysis concerns how the studied methods deal with the problem that artifacts affecting video sequences can be of different intensities. For instance, a video sequence can be much more noisy than another one.

All the image and video restoration approaches can be divided into blind and non-blind methods based on whether they use information about degradation operators or not. On the one hand, non-blind approaches are able to achieve better restoration performance because they can exploit degradation operator information to better understand how to properly remove the artifacts, but they may produce new ones when wrong information is used, as stated by Nah et al. [13]. On the other hand, blind approaches do not have to deal with this problem, but their restoration performance is usually lower than the one achieved by non-blind methods when the degradation operator information they use is correct, as An et al. [52] remarked. There is an open debate on which of the two methods is better.

Using degradation operator information implicitly means that such information is available, but this is not always the case. For example, at training time the σ value of the additive white Gaussian noise may be known because the training samples are synthetically generated. However, such value is unknown when denoising a video sequence at inference time. In this case, non-blind methods cannot be used unless the information they require is estimated by means of external resources.

In the literature, several methods to estimate the parameters of different distortions have been proposed. For instance, Immerkær [53] proposed a simple method to estimate the variance of zero-mean additive white Gaussian noise affecting images, whereas Cogranne [54] proposed an algorithm to estimate the quality factor q of JPEG compressed images based on quantization tables.

Among all the analyzed approaches for video restoration, the only non-blind methods are DVDNet [21] and FastDVDNet [28], as they take advantage of information about the standard deviation of the additive white Gaussian noise affecting the video sequences. In their works, Tassano et al. [21][28] stated that using such information allowed to increase the denoising performance of their networks. Conversely, the other methods do not exploit any information about the intensity of the artifacts.

It is worth noting that some of these methods do not handle different distortion intensities with a single model, meaning that they can restore videos affected by a specific distortion with a specific parameter. For example, MFQE 2.0 [30] and STDF [29] restore videos compressed using the HEVC algorithm with a specific value of the quantization parameter, and restoring a video compressed using a different quantization parameter requires a new model to be trained. Besides, VESPCN [18] and DUF [19] are able to increase the spatial dimension of videos using a specific upscaling factor, requiring different models for different upscaling factors.

In conclusion, it emerged that video restoration frameworks can exploit information about the intensity of the artifacts affecting video sequences to improve restoration performance. If such information is not available, external resources can be adopted to estimate it, but they must be accurate because using wrong information causes the introduction of new artifacts. In addition, some of the analyzed methods do not deal with different distortion intensities using a single model, but they require new models to be trained. However, the vast majority of them are able to restore videos even if the artifacts are of different intensities, using a single model properly trained.

4.1.3 Basic components

The third analysis is related to the basic components of the architectures proposed in the literature. Such basic components include architectural details, what information is used to increase restoration performance, how to perform downscaling and upscaling operations and so on.

From the analysis carried out, the basic components and key ideas behind the architectures under study are the following:

- residual learning: after the introduction of the ResNet architecture [6], residual learning has been widely used in order to ease the training process. It has been shown that learning the transformation that applied to the input produces the output is easier than directly learning the output from the input. Almost all the approaches under study use residual learning, mainly to preserve spatial details and to speed up the training process.

- downscaling and upscaling strategies: there are different ways to perform downscaling and upscaling. The simplest solution is to use traditional interpolation methods, such as bilinear or bicubic. As stated by Shi et al. [9], using such traditional methods to perform upscaling does not carry any additional information to solve the upscaling problem. For this reason, different learnable upscaling methods have been introduced, such as upconvolutions and the sub-pixel convolutional layer [9]. Concerning the downscaling procedure, some methods still use bilinear interpolation, such as EDVR [24], while others adopt strided convolutions, such as DeBlurNet [17] and FastDVDNet [28]. It may be useful to notice that none of them uses max pooling layers, as they are known to lose information [45].

- encoder-decoder architecture: most of the state-of-the-art approaches use an encoder-decoder architecture, in which the encoder extracts significant information from the input by progressively reducing its spatial dimension, while the decoder uses the information extracted by the encoder to construct the output by progressively increasing the spatial dimension of the feature maps. Encoder-decoder networks are widely used in many computer vision tasks, and the most famous architecture is U-Net [55], which was introduced for image segmentation. The video restoration frameworks that use an encoder-decoder architecture are DeBlurNet [17], FastDVDNet [28], STDF [29] and EVRNet [34].

- single-scale architecture: on the one hand, using encoder-decoder architectures allows to learn semantically rich features, but, on the other hand, the downscaling process in such architectures loses fine spatial details, which are difficult to recover in later stages [56]. For this reason, some methods work in the full-resolution domain to produce results with more accurate spatial details. For instance, ViDeNN [23] and MFQE 2.0 [30] use single-scale architectures. However, the main problem is related to the computational cost, as working in the full-resolution domain requires a higher number of operations than the ones required when working at lower resolution.

- end-to-end training: end-to-end learning usually refers to omitting any hand-crafted intermediary algorithms and directly learning the solution of a given problem. In some cases, end-to-end training a neural network has been shown to increase performance. Among the studied video restoration approaches, some of them use specific blocks to perform specific operations. For example, both DVDNet [21] and ViDeNN [23] use spatial and temporal denoisers that are trained sequentially: the output of the spatial denoiser is used to train the temporal denoiser. Tassano et al. [28] stated that, specifically for video denoising, end-to-end training allows to reduce flickering artifacts.

- preprocessing module: when frames are affected by severe distortions, estimating pixel motion becomes a very difficult task. For this reason, some approaches use preprocessing modules in order to improve the quality of the frames before the motion estimation step. For instance, DVDNet [21] and ViDeNN [23] use a specific module for single frame denoising, whereas EDVR [24] uses a pre-deblur module to improve alignment accuracy.

- degradation information exploitation: as already mentioned in Chapter 2, degradation operators are characterized by some parameters. Image and video restoration approaches can be divided into blind and non-blind methods based on whether they use information about degradation operators or not, as remarked in Subsection 4.1.2. DVDNet [21] and FastDVDNet [28] are non-blind methods exploiting information about the intensity of the additive white Gaussian noise (represented by its standard deviation) affecting the video sequences to improve the denoising performance.

- two-stage restoration: two-stage restoration means that the output of the first model is used as input to the same model in a recursive way. Another possibility is to cascade two different models of the same network, so that the output of the first model is used as input to the second one. This is done by FastDVDNet [28], in which the outputs of the first set of denoising blocks are fed to another denoising block. Tassano et al. [28], in their work, studied the impact of the two-stage denoising, confirming that the denoising performance using this strategy increases. This is also confirmed by Wang et al. [24], who showed that feeding the output of the first EDVR [24] model to another EDVR [24] model allows to obtain better restoration performance in all the addressed tasks.

- residual and dense blocks: some architectures among the analyzed ones are inspired by ResNet [6] and DenseNet [57] and use the original or modified blocks taken from the aforementioned networks. Using such blocks allows to mitigate the vanishing gradient problem, to encourage feature reuse and strengthen feature propagation [30]. DeepDeblur [13], DVDNet [21] and EDVR [24] use a cascade of residual blocks [6], whereas DUF [19] and MFQE 2.0 [30] use dense blocks [57].

- temporal and spatial attention: the attention mechanism [27] is used to weight feature maps according to their importance, because not all the features extracted contribute equally to the construction of the restored frame. There are different ways to implement this mechanism. For example, the Squeeze-Excitation block [47] allows to perform inter-channel attention, while the non-local operation [46] is used for spatial attention. EVRNet [34] uses Squeeze-Excitation blocks [47], while EDVR [24] makes use of a specifically designed module, called TSA and described in Subsection 3.2.7, to perform both temporal and spatial attention.
- temporal neighborhood exploitation: temporal information is a key concept for all the video restoration tasks. In order to exploit temporal information, the target frame is fed to the network together with a number of adjacent frames. The dimension of the temporal neighborhood changes based on the architecture. For instance, ViDeNN [23] uses just three frames as input, TOFlow [14] and EDVR [24] use seven frames, while MFQE 2.0 [30] uses the entire video sequence. Although using more frames is expected to produce better results, some studies demonstrated that using too many frames may cause a drop in performance, especially in the case of large motion.

Some considerations can be made about the aforementioned basic components. First, there are no specific modules to perform specific restoration tasks. This is particularly important because it means that an approach designed to address a task may be used to address another task. Second, encoder-decoder architectures are the most used architectures, as they can extract semantically rich features, but their main drawback is that of losing high-frequency components during the downscaling process. On the contrary, using single-scale architectures that do not reduce the spatial dimension allows to preserve spatial details, but they learn less powerful features. Therefore, it may be possible to fuse such architectures to exploit their best characteristics [56]. Third, there is no correct answer regarding the optimal number of adjacent frames to use to produce the best restoration performance, as it depends on the cases and it is based on empirical evaluations. Finally, using distortion information may help networks increase the restoration performance. These considerations should be taken into account to design a new video restoration framework.

4.1.4 Loss functions

The last analysis concerns the loss functions used by the video restoration approaches to understand whether there is any relationship between the used loss function and the addressed task.

Table 4.2: Loss functions used for the training process of the state-of-the-art approaches

Approach name	**Task(s)**	**Loss function**
TOFlow [14]	Super resolution, Denoising, Frame interpolation, Compression artifact removal	L_1
DeBlurNet [17]	Deblurring	MSE
VESPCN [18]	Super resolution	MSE + Huber [50]
DUF [19]	Super resolution	Huber [50]
DVDNet [21]	Denoising	MSE
ViDeNN [23]	Denoising	SSE
EDVR [24]	Super resolution, Denoising, Deblurring, Compression artifact removal	Charbonnier [58]
FastDVDNet [28]	Denoising	MSE
STDF [29]	Compression artifact removal	SSE
MFQE 2.0 [30]	Compression artifact removal	MSE
EVRNet [34]	Super resolution, Denoising, Compression artifact removal	L_1

Table 4.2 reports the loss functions used by the analyzed methods. As shown, most of the approaches use MSE as loss function to deal with all the restoration tasks, except for temporal frame interpolation. L_1 loss is used only by TOFlow [14] and EVRNet [34], while ViDeNN [23] and STDF [29] use SSE instead of MSE.
From this quick analysis it is possible to conclude that the most used loss function is MSE and there is no relationship between a specific loss function and a specific restoration task.

4.2 Selection of the baseline architecture

Although some of the video restoration approaches described in Chapter 3 are able to achieve very good performance in different tasks, the vast majority of them deal with single artifacts. For example, EVRNet [34] achieves good performance in video denoising, super resolution as well as in compression artifact removal. However, the network addresses just a single artifact at a time, and it is not able to restore videos affected by multiple distortions simultaneously. EDVR [24] is effective in restoring multi-distorted videos, but it is not efficient because of its 20 million parameters.

Since the purpose of this thesis is that of creating a deep neural network to restore videos in the presence of multiple artifacts, two routes can be adopted: starting from scratch in developing a new method or extending an existing approach. As starting from scratch is not a good idea for several reasons, such as the limited amount of time available, choosing an existing and promising approach and extending it is the best solution.

Table 4.3: High-level analysis of the state-of-the-art approaches for video restoration used to select the baseline model to take inspiration from

Approach name	**Publication year**	**Task(s)**	**Motion estimation/ compensation**	**Source code**	**Train code**	**Language**
TOFlow [14]	2017	Super resolution, Denoising, Frame interpolation, Compression artifact removal	Explicit	✓	✓	Matlab, Python
DeBlurNet [17]	2017	Deblurring	Implicit	✓	✗	Lua, Matlab
VESPCN [18]	2018	Super resolution	Explicit	✓	✓	Python, Shell
DUF [19]	2018	Super resolution	Implicit	✓	✗	Python
DVDNet [21]	2019	Denoising	Explicit	✓	✗	Python, Shell
ViDeNN [23]	2019	Denoising	Implicit	✓	✓	Python, Shell
EDVR [24]	2019	Super resolution, Denoising, Deblurring, Compression artifact removal	Implicit	✓	✓	Python, C++, Cuda, Matlab
FastDVDNet [28]	2020	Denoising	Implicit	✓	✓	Python, Shell
STDF [29]	2020	Compression artifact removal	Implicit	✓	✓	Python, C++, Cuda, Shell
MFQE 2.0 [30]	2020	Compression artifact removal	Explicit	✓	✓	Python
EVRNet [34]	2020	Super resolution, Denoising, Compression artifact removal	Implicit	✗	✗	-

There are some critical aspects that must be taken into account in order to decide which of the analyzed approaches should be used as baseline model to take inspiration from. The availability of the source code and the training code is an essential aspect, as some works accurately describe the proposed approach without providing the code to run it. All the analyzed methods are provided with the source code, except for EVRNet [34]. Unfortunately, DeBlurNet [17], DUF [19] and DVDNet [21] are not provided with the train code, therefore they cannot be extended to different tasks because it is not possible to train new models. Another critical aspect is the method used to estimate pixel motion and perform frame alignment, as it may affect the restoration performance. The advantages and disadvantages of these approaches have already been discussed in Subsection 4.1.1. Based on those observations, using methods that perform implicit pixel motion estimation and frame alignment appears the most suitable solution because explicitly estimating pixel motion in the case of multi-distorted videos could be very challenging and, thus, motion estimation errors may be propagated throughout the entire network producing additional artifacts. TOFlow [14], VESPCN [18], DVDNet [21] and MFQE 2.0 [30] explicitly estimate pixel motion and perform frame alignment, while DeBlurNet [17], DUF [19], ViDeNN [23], EDVR [24], FastDVDNet

[28], STDF [29] and EVRNet [34] do them implicitly.
All the results of this analysis are summarized in Table 4.3. Based on these considerations, the methods whose code is not available or use an explicit motion estimation and compensation steps have been discarded. The implementation of the remaining methods has been analyzed more in detail, starting from the most recent one.
The first method analyzed was STDF [29], a state-of-the-art approach to remove compression artifacts from videos proposed by Deng et al. in 2020. Implemented in Python using Pytorch [59], it uses implicit motion estimation and compensation and it is end-to-end trainable. Unfortunately, the code of STDF [29] does not work "as-is" even after having allocated three hours for environment setup due to its external dependencies and, hence, it has been discarded.
The second method analyzed was FastDVDNet [28], a state-of-the-art approach for video denoising proposed by Tassano et al. in 2020, implemented in Python using Pytorch [59]. The source code is available, well documented and it is easy to read and understand. In addition, also the code for training models is available. Motion estimation and compensation are implicitly performed by the network and learned during the training process. The network is end-to-end trainable and is very efficient, achieving real-time video denoising performance. Finally, the source code provided by the authors works without any further intervention, allowing both to train a new model and to test pre-trained models.
For these reasons, in addition to other interesting characteristics of the network, FastDVDNet [28] has been selected as baseline model on which to build a new deep neural network able to restore videos affected by multiple artifacts.

4.3 FastDVDNet in detail

In order to fully comprehend how FastDVDNet [28] works, a further analysis aiming to investigate the main characteristics of this network is necessary. In particular, the most interesting ones are related to the structure of the architecture and the way it exploits information about distortion parameters to improve the denoising performance.

4.3.1 Spatio-temporal information

Temporal coherence and flickering removal are fundamental aspects for each video denoiser. In order to achieve these, any video denoiser should use temporal information existing in neighboring frames.
The architecture of FastDVDNet [28] has been designed to be able to extract both spatial and temporal information from the target frame and its neighboring frames. Using spatial and temporal information means that, when denoising a given pixel, the network can look for similar pixels not only in the target frame but also in adjacent ones. For this reason, it is important to provide the network also with adjacent frames in order to fully exploit temporal redundancy, which is useful to

extract meaningful features to effectively remove noise from videos. To this end, FastDVDNet [28] takes five consecutive frames as input, considering the central frame as the target. Given the target frame at time t, f_t, its four neighboring frames are stacked and fed to the network, that is, the input of the network is $\{f_{t-2}, f_{t-1}, f_t, f_{t+1}, f_{t+2}\}$.

4.3.2 Motion handling

As already mentioned in Section 4.2, one of the main reasons why FastDVDNet [28] has been selected among other approaches is its ability to implicitly handle pixel motion between frames.

Especially for video denoising, explicitly handling motion adds an additional complexity, as pixel values are not constant over time. This makes it very difficult to estimate how pixels have moved when dealing with severe noisy frames. Wrong motion estimation, which is usually performed at the beginning of every video restoration framework, implies wrong alignment, i.e. the neighboring frames are not correctly warped to the target one, leading to visible artifacts because the extracted information are not spatially precise. This is why FastDVDNet [28] does not include any specific module for motion estimation and frame alignment, which are implicitly embedded within the network itself, thus avoiding the problem of introducing new artifacts because of wrong motion estimations.

4.3.3 Noise map

An important characteristic of all the state-of-the-art approaches to both image and video restoration is that of being either blind or non-blind methods. Blind methods perform restoration without any information about the distortion affecting the image or the video, while non-blind methods require some information about the distortion to obtain optimal restoration results. In the latter case, there are different ways to provide the network with degradation operator information. For example, An et al. [52], in their work related to single image deblurring, estimate the blur kernel, encode it and feed it to the network alongside the blurred image. FastDVDNet [28] performs non-blind video denoising, as it requires additional information about the noise intensity. In addition to a noisy video sequence, it takes a noise map as input, which is nothing but an additional channel that, concatenated to the noisy frames, is fed to the network to provide useful information that the network can exploit to understand the severity of the distortion and consequently remove it. In this case, since the considered noise is uniform zero-mean additive white Gaussian noise, the noise map is filled with the value of the standard deviation of the noise affecting the video sequence.

Tassano et al. [28] also stated that such noise map could be very useful in the case of spatially variant noise. However, in their work, they trained FastDVDNet [28] only using uniform noise and they did not provide any experimental result about spatially variant noise.

It is worthwhile to mention that such noise map is also used at inference time. However, at inference time, the actual standard deviation of the noise corrupting the video sequence is usually unknown. For this reason, the network as delivered by its authors cannot be used in real cases unless an additional method to estimate the standard deviation of the noise affecting the video sequence is devised.

4.3.4 Denoising block

The main component of FastDVDNet [28] is the denoising block, which is a modified U-Net [55] architecture. U-Net [55] is an encoder-decoder architecture with skip connections that forward the output of each encoder layer directly to the input of the corresponding decoder layer.
In contrast to the original U-Net [55] architecture, Tassano et al. [28] modified the denoising block as follows:

- the encoder is adapted to take three adjacent frames and the noise map as input;
- the upscaling in the decoder is performed using Pixel Shuffle layers [9], which are used to reduce gridding artifacts;
- skip connections apply pixel-wise addition instead of channel-wise concatenation, reducing memory requirements;
- residual learning [6] is applied by subtracting the output of the block from the input central frame, making the training process easier.

Figure 3.17 in Subsection 3.2.8 illustrates the architecture of the denoising block. More in detail, three consecutive frames and the noise map are stacked and given to the denoising block as input. The encoder is composed of eight convolutional layers, which are followed by batch normalization [8] and ReLU [60]. The downscaling operation is performed using strided convolutions, and the channel depth varies from 32 up to 256. The encoder halves the spatial dimension of the input frames twice so that, at the bottleneck, the spatial dimension of the feature maps corresponds to a quarter of the original spatial dimension. The decoder is composed of eight convolutional layers as well, but the channel depth varies from 256 down to 3, corresponding to the channels of the restored frame in the RGB domain. As already mentioned, upscaling is performed using Pixel Shuffle layers [9] in order to reduce gridding artifacts. Finally, residual learning [6] is applied by subtracting the output of the decoder from the central noisy input frame.

4.3.5 Two-stage denoising

FastDVDNet [28] is a two-step cascaded architecture. As shown in Figure 3.16, the first stage consists of three parallel denoising blocks, whose weights are shared, while the second stage is performed by an additional denoising block. Each denoising block in the first stage takes three neighboring frames, along with the noise map, as input and tries to recover the central frame. Then, the outputs of the blocks of the first stage are stacked and, along with the noise map, are fed to the denoising block of the second stage to produce the final result.

The decision of such architecture is motivated by the fact that, in this way, the network can effectively employ information from the temporal neighbors and enforce the temporal correlation of the remaining noise in the output frame.

Tassano et al. [28] provided a demonstration that two-stage denoising outperforms single-stage denoising. They compared the two-step cascaded architecture of FastDVDNet [28] with a single-step architecture using more frames as input, concluding that the former is able to obtain better performance because the latter showed a sharp increase of flickering artifacts.

4.3.6 Training details

The training process of FastDVDNet [28] is supervised, therefore, it needs pairs of input-output frames.

The network has been trained using 384000 patches randomly extracted from the DAVIS 2017 dataset [61], to which additive white Gaussian noise with variable σ is added. More in detail, given a set of five consecutive frames, five patches of size 96×96 are randomly cropped at the same spatial location, one for each frame. Then, additive white Gaussian noise with random $\sigma \in [5, 55]$ is added to the cropped patches. The ground truth is given by the patch cropped from the central frame (the third frame, in this case) without any noise applied. The noise map is a single channel of size 96×96 filled with the value of the used σ. The five 96×96 patches along with the noise map compose a training sample. Note that using patches of size 96×96 is sufficient to provide enough overlapping contents in the stack even if the frames are not aligned. Data augmentation is performed using random horizontal and vertical flips.

MSE is used as loss function between the input frame and the corresponding ground truth using Adam [62] as optimizer. The training process proceeds for 80 epochs using mini-batches of size 96. Regarding the learning rate, the initial learning rate is set to $1e^{-3}$ for the first 50 epochs, then changes to $1e^{-4}$ for the following 10 epochs and then changes again to $1e^{-6}$ for the remaining epochs.

4.3.7 Inference

In order to use FastDVDNet [28] to remove noise from videos, each frame must be surrounded by other four frames representing its temporal neighborhood. One may think which neighboring frames can be used, for example, for the first frame of the video sequence, as it does not have any previous frame. A possible solution may be that of using its consecutive frames (referring to the previous example, from the second frame to the fifth one). However, the network is trained just to recover the central frame, so the aforementioned solution is not feasible. For this reason, if the natural temporal neighborhood is not available, some previous and subsequent frames are used to wrap the target one so that it is placed in the central position. For instance, given a five-frame video sequence, the frames are concatenated as follows:

- to restore the first frame, f_1: $[f_3, f_2, f_1, f_2, f_3]$
- to restore the second frame, f_2: $[f_2, f_1, f_2, f_3, f_4]$
- to restore the third frame, f_3: $[f_1, f_2, f_3, f_4, f_5]$
- to restore the fourth frame, f_4: $[f_2, f_3, f_4, f_5, f_4]$
- to restore the fifth frame, f_5: $[f_3, f_4, f_5, f_4, f_3]$

In this way, the target frame is always placed in the center position, and the information coming from its temporal neighborhood can be used to restore it.

4.4 Proposed method for video restoration: MdVRNet

In order to accomplish this thesis goal, that is, designing a deep neural network to restore videos affected by multiple distortions, the denoising capability of FastDVDNet [28] and its potential flexibility to be adapted to other restoration tasks have been exploited and further improved by using the insights identified from the in-depth analysis carried out in Section 4.1. The result is a new multi-distorted video restoration network able to restore videos even when they are corrupted by multiple degradation operators. This network is called Multi-distorted Video Restoration Network, abbreviated MdVRNet.

4.4.1 Architecture overview

The MdVRNet framework takes inspiration from the analyzed FastDVDNet [28], which is a two-step cascaded architecture taking five consecutive frames as input and using a noise map to obtain additional information about the noise intensity to increase the denoising performance.

MdVRNet exploits an original distortion parameter estimation module properly devised to obtain information about degradation operators affecting video sequences and make the restoration process more robust, and a novel multi-scale restoration block similar to the denoising block of FastDVDNet [28] but with the following improvements: the additional full-resolution feature branch, used to extract features at full resolution in order to learn fine pixel dependencies and avoid losing details, and the channel attention mechanism used to weight the features extracted by the low-resolution and the full-resolution feature branches, according to the importance they have in reconstructing the target frame.

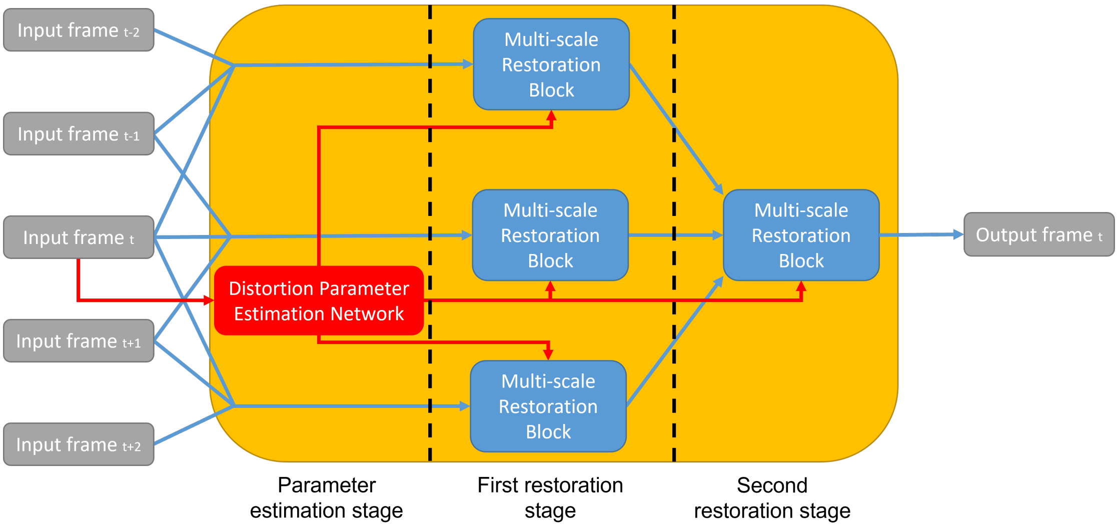

Input frame t-2
Input frame t-1
Input frame t
Input frame t+1
Input frame t+2
Distortion Parameter Estimation Network
Multi-scale Restoration Block
Multi-scale Restoration Block
Multi-scale Restoration Block
Multi-scale Restoration Block
Output frame t
Parameter estimation stage
First restoration stage
Second restoration stage

Figure 4.1: MdVRNet architecture proposed to restore videos affected by multiple distortions

An overview of the framework is shown in Figure 4.1. The target frame is given to the distortion parameter estimation module to estimate the degradation parameters required by MdVRNet, and the extracted information is propagated to each multi-scale restoration block to make them aware about the intensity of the artifacts affecting the video sequence, increasing the restoration performance. Each group of three consecutive frames is fed to the corresponding multi-scale restoration block composing the first restoration stage, which contains three blocks in total. The outputs of the first restoration stage are used as inputs to the second restoration stage, which consists of just a single multi-scale restoration block. The output of the second restoration stage corresponds to the restored frame. Overall, the MdVRNet framework contains about 3 million parameters.
In summary, the basic components of MdVRNet are the following:

- noise map estimation
- multi-scale restoration block
- two-stage restoration
- implicit motion estimation and frame alignment

4.4.2 Noise map estimation

As mentioned in Subsection 4.3.3, in the original version of FastDVDNet [28] proposed by Tassano et al. [28] the noise map must be filled with the true value of the σ parameter of the additive white Gaussian noise affecting the video sequence. However, such value may be unknown at inference time and an additional resource to estimate it is needed.

It is important to notice that MdVRNet is designed to restore multi-distorted videos, therefore, such additional resource must be able to extract information related to multiple distortions. Different methods to estimate the distortion parameters of different degradation operators have been mentioned in Subsection 4.1.2. However, although they can accurately estimate the considered distortion parameters, they are designed to estimate these parameters only in the presence of single artifacts and they may produce inaccurate estimations when multiple distortions are present.

To address this problem, a new CNN called Distortion Parameter Estimation Network and dubbed DPEN has been devised. DPEN is a feedforward network consisting of five convolutional blocks and three fully connected blocks, as shown in Figure 4.2. Concerning the convolutional component of DPEN, each convolutional block is composed of a convolutional layer followed by ReLU [60], batch normalization [8] and max pooling, except for the fourth and the fifth convolutional blocks in which batch normalization [8] has been removed. The kernel size is set to five in the first three convolutional layers, to three in the fourth convolutional layer and to one in the last one. The depth varies from 8 up to 128 and the spatial dimension is halved by each max pooling layer so that the output shape of the last convolutional block corresponds to a fifth of the input shape. The downscaling operation is performed just by max pooling layers, as the convolutional layers use zero-padding to preserve the spatial dimension. Regarding the fully connected component, the first fully connected layer has 64 neurons, the second one has 32 neurons and the last one has two neurons, which output the estimated values of the distortion parameters. Each fully connected layer is followed by ReLU [60] except for the last one, which uses Sigmoid to compress the output value in the $[0, 1]$ range. Finally, the convolutional and fully connected components of DPEN are connected by means of a global average pooling layer, so that there is no constraint about the shape of the input images.

It is important to notice that DPEN outputs global values because it is devised under the assumption of globally distributed artifacts, that is, the artifact intensity is constant in all the spatial positions of a given frame.

In the case of video denoising, the additive white Gaussian noise is characterized by two parameters: the mean and the standard deviation. Since the mean is always considered as zero, the only parameter to estimate is the value of the standard deviation. Also in the case of video deblurring, Gaussian kernels are characterized by two parameters: the dimension of the kernel and the standard deviation. Since the kernel size depends on the standard deviation, only this latter parameter needs to be estimated. In the case of JPEG compression, the parameter to estimate is the value of the quality factor q. Estimating such parameters is equivalent to estimating the intensity of the artifacts, as there is a correlation between them: the higher the value of σ, the higher the intensity of noise or blur; the lower the value of q, the higher the intensity of the blocking artifacts. For these reasons, the network outputs two values depending on the artifacts it has to deal with.

One may notice that DPEN takes one frame as input instead of a frame sequence.

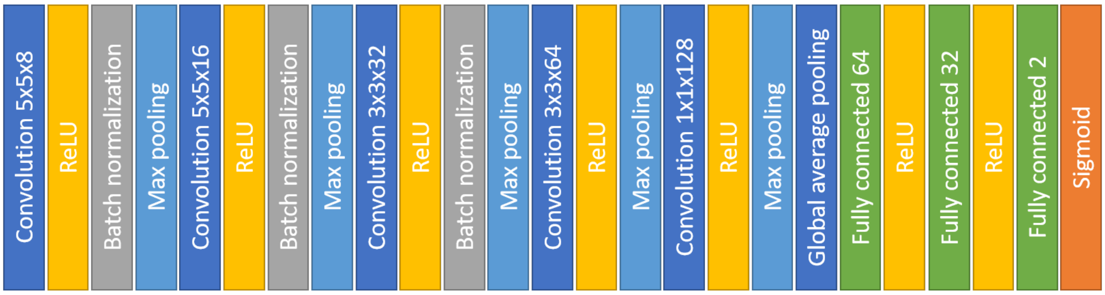


Figure 4.2: Architecture of the neural network devised to estimate distortion parameters, named DPEN

This because it is assumed that all the frames of a video sequence contain the same distortion intensity, thus using a frame sequence instead of a single frame would not bring additional information.

DPEN is a very shallow network, as it has just about 53K parameters. Therefore, it can be used to estimate the distortion parameters required by MdVRNet introducing very little overhead.

Since only combinations of two distortions have been considered in this thesis, as mentioned in Section 2.4, the noise map used by MdVRNet contains two channels, so that the first one can be filled with the value of the first distortion parameter, such as the σ value of additive white Gaussian noise, whereas the second one can be filled with the second distortion parameter, i.e. the quality factor q used to compress frames.

4.4.3 Multi-scale restoration block

The effectiveness of MdVRNet in restoring multi-distorted videos lies on the multi-scale restoration block, which is a two-stream network that allows to extract spatial and temporal features at different scales, weight them according to their importance using an attention mechanism and fuse them to obtain a degradation map that is finally removed from the degraded target frame.

The detailed representation of the multi-scale restoration block is shown in Figure 4.3. A stack of three consecutive frames along with the noise map estimated by DPEN are used as input. After a set of two convolutions, each of which followed by batch normalization [8] and ReLU [60], the computation is broken into two parallel branches working at different resolutions.

The first branch works at full resolution in order to extract fine pixel dependencies, capturing spatially accurate details. This branch is important to restore the target frame without losing high-frequency components, such as edges. The first convolutional layer is used to increase the number of feature maps from 32 to 64. Then, a set of three residual blocks are applied in order to learn the degradation map at full resolution, paying more attention to finer details. The residual block used by MdVRNet is very similar to the residual block used by DeepDeblur [13]. Finally, the number of feature maps are reduced from 64 back to 32 using a final convolutional layer. The full-resolution branch contains a total of 8 convolutional layers, which are able to extract useful information without increasing the computational cost too much.

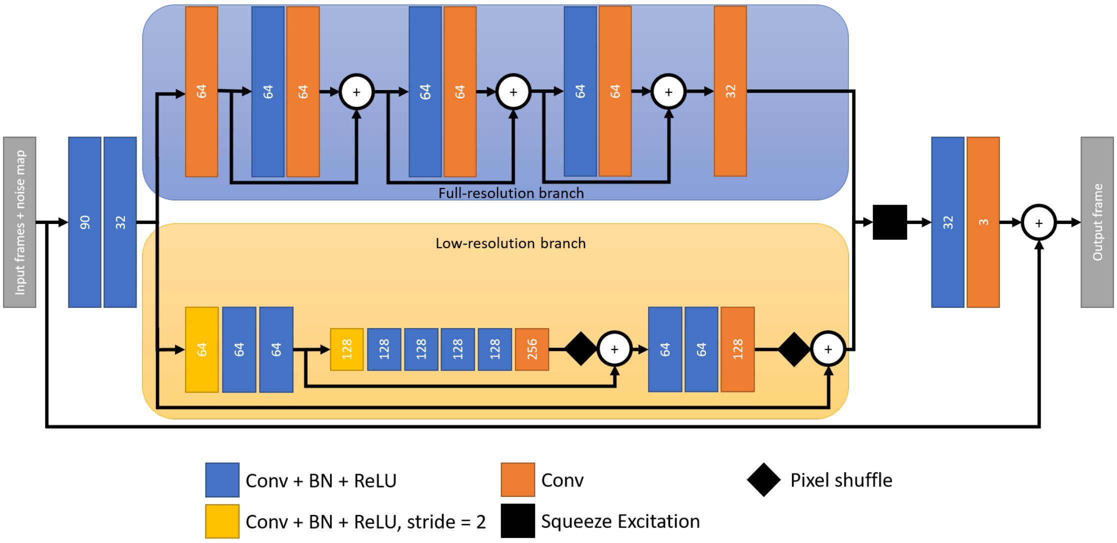


Figure 4.3: Multi-scale restoration block used by MdVRNet to restore multi-distorted videos

The second branch allows to extract coarse pixel dependencies in local areas to obtain semantically rich features using an encoder-decoder architecture working at low resolution. Downsampling is performed using strided convolutions, each one halving the spatial dimension. There are a total of two downscaling operations so that, at the bottleneck, the spatial dimension corresponds to a quarter of the input spatial dimension. As the input passes through this branch, a set of convolutional layers, batch normalization [8] and ReLU [60] decreases the spatial resolution while increasing the number of feature maps. Skip connections forward the output of each encoder layer directly to the input of the corresponding decoder layer using pixel-wise addition to ease and speed up the training process. Upsampling is performed using Pixel Shuffle layers [9] to reduce gridding artifacts.
The features extracted by the two branches are then concatenated and passed through a Squeeze-Excitation block [47], which performs channel attention to weight each feature map according to its importance in reconstructing the target frame. The detailed representation of the Squeeze-Excitation block [47] is shown in Figure 4.4.
The weighted feature maps are then fused together using a final set of convolutional layers, batch normalization [8] and ReLU [60] to obtain the degradation map, which is finally subtracted from the degraded target frame to remove the artifacts and consequently restore it. It is worthwhile to point out that the degradation map is composed of three channels, each of which contains the artifacts detected by the network within each RGB channel of the degraded target frame.

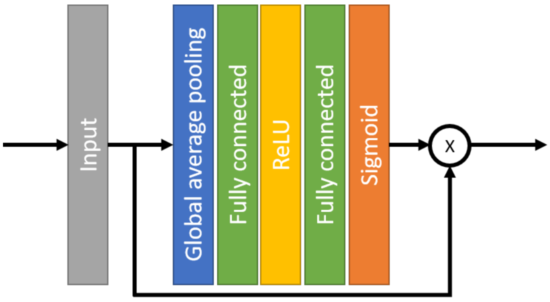


Figure 4.4: Squeeze-Excitation block [47] used by MdVRNet to weight the features extracted by the two branches of the multi-scale restoration block

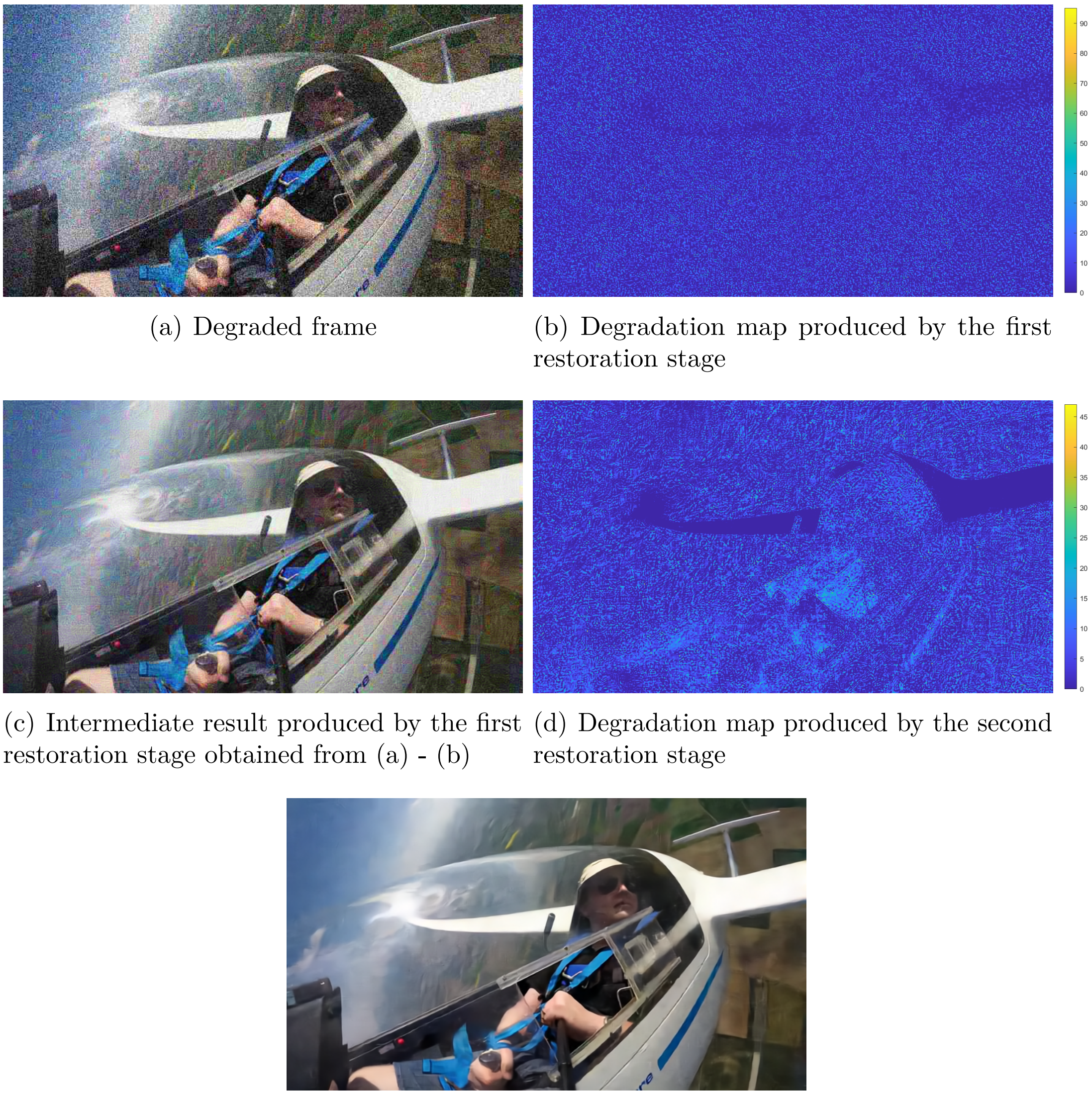
4.4.4 Two-stage restoration

In their work, Tassano et al. [28] have shown that a two-stage network outperforms a single-stage network in removing noise from videos. This is why FastDVDNet [28] is a two-step cascaded architecture. This characteristic is also inherited by MdVRNet.

The restoration process is split into two stages: the first stage aims to provide intermediate results that will be used by the second stage to produce the final result. The first restoration stage is composed of three multi-scale restoration blocks placed in parallel. Each block takes a stack of three consecutive frames, together with the noise map estimated by DPEN, as input. Note that the weights of the three multi-scale restoration blocks of the first stage are shared, that is, all the blocks perform the same identical operations. The second restoration stage is composed of just a single multi-scale restoration block, which takes the outputs of the first stage and the noise map estimated by DPEN as input.

Ideally, the first restoration stage should pay more attention to single pixel restoration using information coming from very close neighboring pixels both in the same frame and in the closest frames, that is, the previous and the subsequent one. This because the multi-scale restoration blocks within the first stage use short-term information and do not have access to long-term information, as each of them sees just a limited temporal neighborhood, i.e. only the previous and the subsequent frame of the given frame. Instead, the second restoration stage should pay more attention to restoring local areas, since the multi-scale restoration block of this stage has access to long-term information and it has a more complete vision of the scene. This because it takes the three output frames of the first restoration stage as input that, in turn, contain information extracted from all the five frames corresponding to the input of MdVRNet.

In order to support the aforementioned statements, Figure 4.5 shows an example of a degraded video frame, obtained by initially adding noise and then compressing it using the JPEG algorithm, restored by MdVRNet. The figure also shows the degradation map and the intermediate result produced by the first restoration stage, as well as the degradation map produced by the second restoration stage. More in detail, Figure 4.5(b) shows the degradation map produced by the first



(a) Degraded frame

(b) Degradation map produced by the first restoration stage

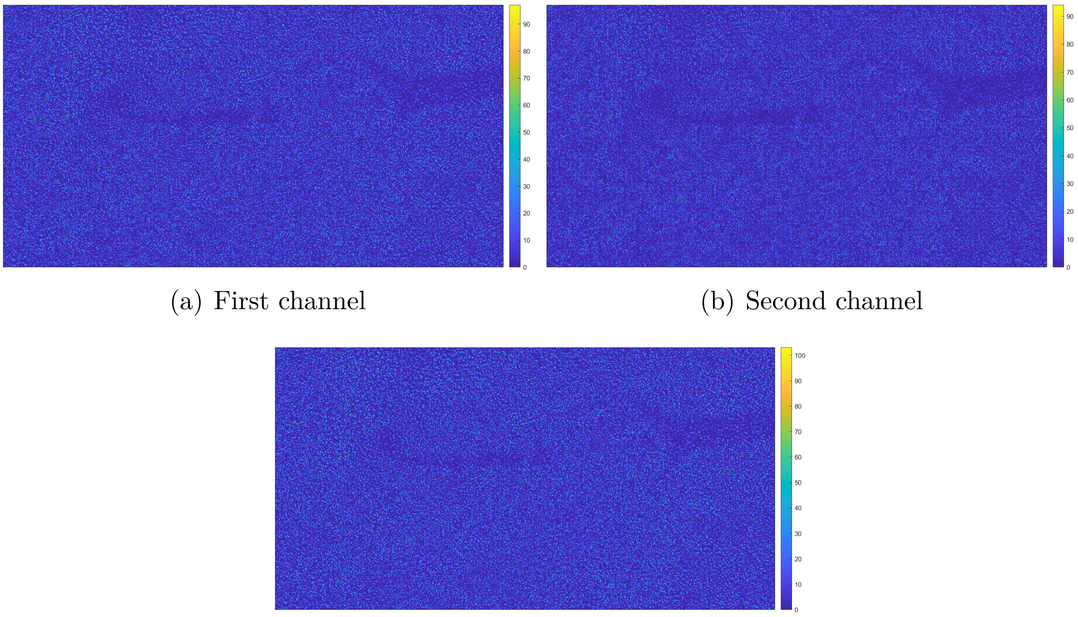
(c) Intermediate result produced by the first restoration stage obtained from (a) - (b)

(d) Degradation map produced by the second restoration stage

(e) Restored frame obtained from (c) - (d)

Figure 4.5: Example of frames and degradation maps extracted from the restoration process of MdVRNet, showing the artifacts removed by each restoration stage. The color of the degradation maps has been modified for a better interpretation.

restoration stage, which subtracted from the degraded frame in Figure 4.5(a) allows to obtain the intermediate result shown in Figure 4.5(c). As shown, the degradation map in Figure 4.5(b) contains fine artifacts at pixel level, which seem to correspond to noise. Figure 4.6 reports the degradation map produced by the first restoration stage divided into channels, allowing to better see its content. Moving forward in the restoration process, the second restoration stage produces the degraded map illustrated in Figure 4.5(d), whose artifacts are clearly different from the ones in Figure 4.5(b). Indeed, they seem coarser artifacts related to JPEG, which acts on local areas of size 8 × 8 rather than single pixels. Interestingly, the artifacts on the

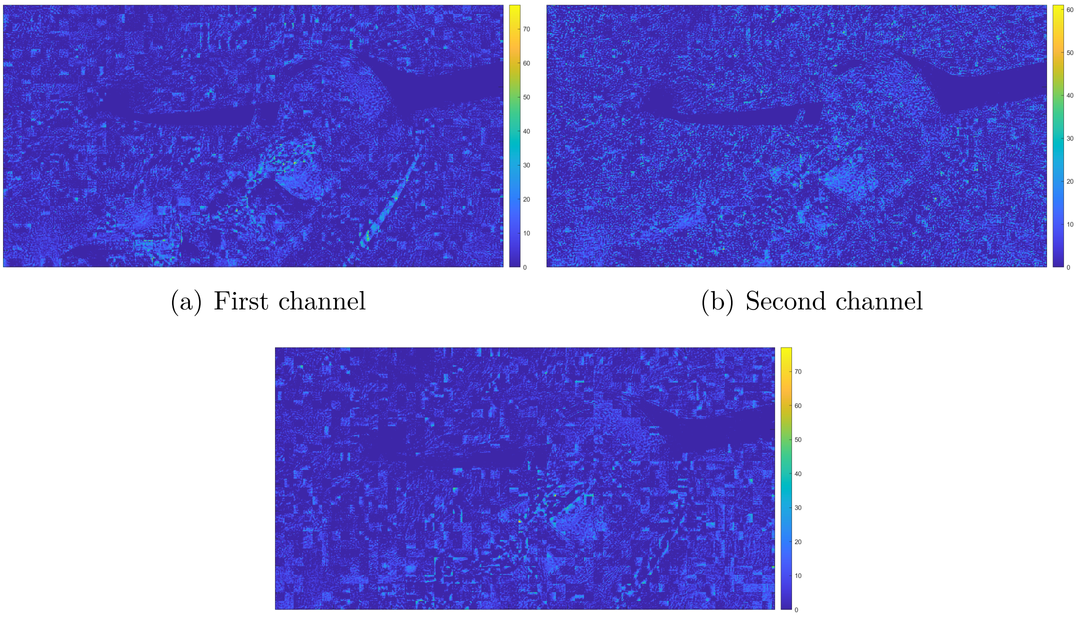


(a) First channel

(b) Second channel

(c) Third channel

Figure 4.6: Channels of the degradation map reported in Figure 4.5(b)



(a) First channel

(b) Second channel

(c) Third channel

Figure 4.7: Channels of the degradation map reported in Figure 4.5(d)

wings of the plane are very mild, meaning that they have been almost completely removed by the first restoration stage. Since the wings are uniform areas, the network treats the entire region in the same way. Figure 4.7 reports the degradation map produced by the second restoration stage decomposed into channels, in which the JPEG related artifacts are more evident. Finally, the artifacts affecting the intermediate result are removed by subtracting the degradation map of the second restoration stage in Figure 4.5(d), producing the restored frame shown in Figure 4.5(e).

4.4.5 Implicit motion estimation and alignment

Pixel motion estimation and frame alignment are tasks that all the video restoration approaches should fulfill to effectively exploit spatial and temporal information coming from both the target frame and its adjacent ones in order to avoid flickering. As already discussed in Subsection 4.1.1, pixel motion estimation is a difficult task even when videos are of high quality, and it becomes more difficult when videos contain some kind of artifacts. Furthermore, when multiple artifacts are present in a video sequence, the correlation among the values of the same pixel in adjacent frames may be broken, making the motion estimation process even more challenging. This is the reason why explicitly estimating pixel motion is not suitable when videos are corrupted by multiple distortions.

It is important to point out that, given five consecutive frames as input, pixel motion estimation and frame alignment between the target frame and its neighboring frames are not computed in a single step. This because, in the case of large motion, it may be difficult to align the target frame and a frame that is not its immediate neighbor, that is, it is neither the previous nor the subsequent frame. This is why there are three multi-scale restoration blocks within the first restoration stage, each of which takes just three consecutive frames as input instead of the entire temporal neighborhood of the target frame. In this way, each multi-scale restoration block can focus on searching for similar pixels in a smaller neighborhood, better employing temporal information.

Chapter 5

Experimental results

This chapter describes the experiments carried out to evaluate the effectiveness of FastDVDNet [28] in removing both single and multiple artifacts from videos, the performance of DPEN in predicting the degradation operator parameters and the restoration performance of the proposed MdVRNet when dealing with multi-distorted videos, bringing out interesting considerations based on the obtained results.

More in detail, the dataset and the metrics used to conduct the evaluation and assess the effectiveness of the networks in restoring distorted videos are introduced in Section 5.1 and 5.2, respectively, while Section 5.3 reports the training details used to train all the different models of FastDVDNet [28] and the proposed MdVRNet. In Section 5.4, the flexibility of FastDVDNet [28] in removing different but single artifacts from videos is evaluated, while in Section 5.5 the network is assessed on the task of multi-distorted video restoration, when videos are affected by multiple distortions at the same time. The experiments in Section 5.6 aim to evaluate the accuracy of DPEN in predicting degradation operator parameters and its attitude in being integrated into existing blind methods. Finally, Section 5.7 investigates the performance of the proposed MdVRNet in restoring videos affected by noise and compression artifacts, focusing on the contribution its basic components give.

5.1 Dataset

The Densely-Annotated VIdeo Segmentation (DAVIS) 2017 dataset was created by Pont-Tuset et al. [61] for the 2017 DAVIS challenge on video object segmentation. It contains 120 video sequences representing both indoor and outdoor scenes. Each sequence is composed of a variable number of frames, ranging from 25 to 127, for a total of 10459, and each frame is an RGB image of size 480×854 in JPEG format. The dataset is divided into a training set, containing 90 sequences, and a test set, containing the remaining 30 sequences. All the experiments in this chapter have been conducted using this dataset, which can be downloaded at `https://davischallenge.org/index.html`.

5.2 Evaluation metrics

In order to evaluate the effectiveness of both FastDVDNet [28] and the proposed MdVRNet in removing artifacts from videos, a set of objective criteria is needed. There are several metrics proposed to assess the quality of restoration methods. Among the available metrics, the results obtained in all the experiments have been quantitatively assessed in terms of peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [12].

5.2.1 Peak signal-to-noise ratio

Peak signal-to-noise ratio (PSNR) is one of the main full-reference metrics used to measure the quality of reconstruction algorithms. It is defined as the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Since many signals have a very wide dynamic range, PSNR is usually expressed as a logarithmic quantity using the decibel scale.

PSNR is defined via MSE. When dealing with images, the mean squared error (MSE) allows to compare the true pixel values of the original image with the ones of the degraded image. It represents the average of the squares of the errors between the original image and the noisy image, where the error is the amount by which the values of the degraded image differ from the original ones. For color images, MSE is computed over all pixel values of each individual channel and it is averaged with the number of color channels.

Mathematically, given two images I and K of size $n \times m$, where I is the original image and K is its noisy approximation, MSE is computed as follows:

$$MSE(I, K) = \frac{1}{n \times m} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} (I_{i,j} - K_{i,j})^2 \tag{5.1}$$

Given the MSE between I and K, PSNR is computed as follows:

$$PSNR(I, K) = 20 \cdot \log_{10} \frac{MAX_I}{\sqrt{MSE(I, K)}} \tag{5.2}$$

where MAX_I is the maximum pixel value of the image I. Since MSE measures pixel errors and a low value of MSE implies a higher value of PSNR, the higher the PSNR, the better. Note that, when the compared images are identical, MSE is 0 and PSNR is undefined (division by zero).

Although PSNR is a widely used metric to assess the effectiveness of reconstruction algorithms, it strictly relies on numeric comparison without taking into account any perceptual factors of the human vision system.

5.2.2 Structural similarity index

Structural similarity index (SSIM) [12] is another widely used full-reference method for measuring the similarity between two images. In contrast to PSNR, SSIM [12] is a perception-based metric that considers image degradation as perceived change in structural information, incorporating important perceptual phenomena including both luminance masking and contrast masking terms. Structural information is the idea that pixels have strong inter-dependencies, especially when they are spatially close. These dependencies carry important information about the structure of the objects in the visual scene. Luminance masking is a phenomenon whereby image distortions tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or texture in the image.

Instead of using traditional error summation methods, SSIM [12] models image distortion as a combination of three factors: luminance distortion, contrast distortion and structural distortion.

Given two images X and Y of the same size, SSIM [12] is computed as follows:

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{5.3}$$

where μ_x and μ_y are the average pixel values, σ_x^2 and σ_y^2 are the pixel variances and σ_{xy} is the pixel covariance of X and Y. Moreover, c_1 and c_2 are two variables used to stabilize the division when the denominator is close to zero, computed as k_1L and k_2L, respectively. L is the dynamic range of pixel values (usually 255), k_1 is equal to 0.01 and k_2 is equal to 0.03 by default.

In contrast to PSNR, which takes values in the $[0, \infty]$ range, SSIM [12] assumes values in the $[0, 1]$ range. Also in this case, the higher the SSIM [12], the better. When the compared images are identical, SSIM [12] is equal to 1.

5.3 Training details

As mentioned in Subsection 4.3.6, Tassano et al. [28] trained their FastDVDNet [28] for 80 epochs with 384000 training samples each. However, training the network with this configuration for each experiment would require a huge amount of time. Therefore, due to the limited time budget, all the following experiments have been conducted training the models for 8 epochs with 256000 training samples each, and a mini-batch size set to 32 for a total of 64000 steps. The learning rate has been set to $1e^{-3}$ for the first five epochs, and to $1e^{-4}$ for the remaining three. To reduce the time required to train the models, patches of size 64 × 64 have been used. Adam [62] has been adopted as optimizer and MSE as loss function, according to the original training setup.

During the experiments it emerged that the used configuration was enough to converge, that is, the training loss no longer decreased even lowering the learning rate. This allows to use the obtained results to draw unbiased conclusions about the effectiveness of the networks in removing both single and multiple artifacts from videos, as well as on the impact that providing the networks with the noise map or not has on the restoration performance.

Concerning the use of the noise map for training, since the architectures of FastDVDNet [28] and MdVRNet are designed to take a fixed number of channels as input, the non-blind restoration has been performed by filling the noise map with the distortion parameters used to generate the training samples, whereas the blind restoration by filling the noise map with zeros: in the former case the networks receive information about the intensity of the distortions affecting the input frames, in the latter case they do not.

5.4 FastDVDNet on single artifacts

The first step towards the extension of FastDVDNet [28] to restore videos affected by multiple artifacts is that of evaluating its flexibility in restoring videos corrupted by different but single distortions. To this end, several experiments have been conducted on videos affected by single distortions, i.e. noise, blur and compression artifacts. The motivation behind these experiments is that, if the network is not able to restore single-distorted videos, it will not be able to restore them when multiple distortions are present.

Besides, given that Tassano et al. [28] did not provide any evaluation concerning how the results change whether the noise map is used or not, these experiments also have the purpose of assessing this aspect. For simplicity reasons, the model trained using the noise map is called “non-blind”, as it receives information about degradation operators, whereas the model trained without the noise map is called “blind”, as it does not know the intensity of the artifacts affecting the video sequence.

5.4.1 Denoising

The first experiment aims to evaluate how the denoising performance of FastDVDNet [28] changes whether information about degradation operators is used or not. In fact, although the network is designed to remove noise from videos, only the results obtained by the non-blind model are available. To perform this comparison, two new models, one using the noise map and one not, have been trained using the same training configuration and evaluated.

The training samples have been generated by adding additive white Gaussian noise with σ randomly extracted from the $[5, 55]$ range to the clean frames. Note that, given a sample consisting of five consecutive frames, the added noise has the same value of σ, meaning that every frame within a training sample has the same noise intensity. Moreover, since the pixel values after adding noise may exceed the $[0, 255]$ range, all the values outside such range have been clipped.

The denoising performance of FastDVDNet [28] is reported in Figure 5.1. As expected, the performance obtained by the non-blind model is better than the one obtained by the blind model. This is due to the fact that, in the latter case, the network does not know the level of noise affecting the video sequence. Indeed, the network removes some noise from the noisy frames, but it is unable to completely

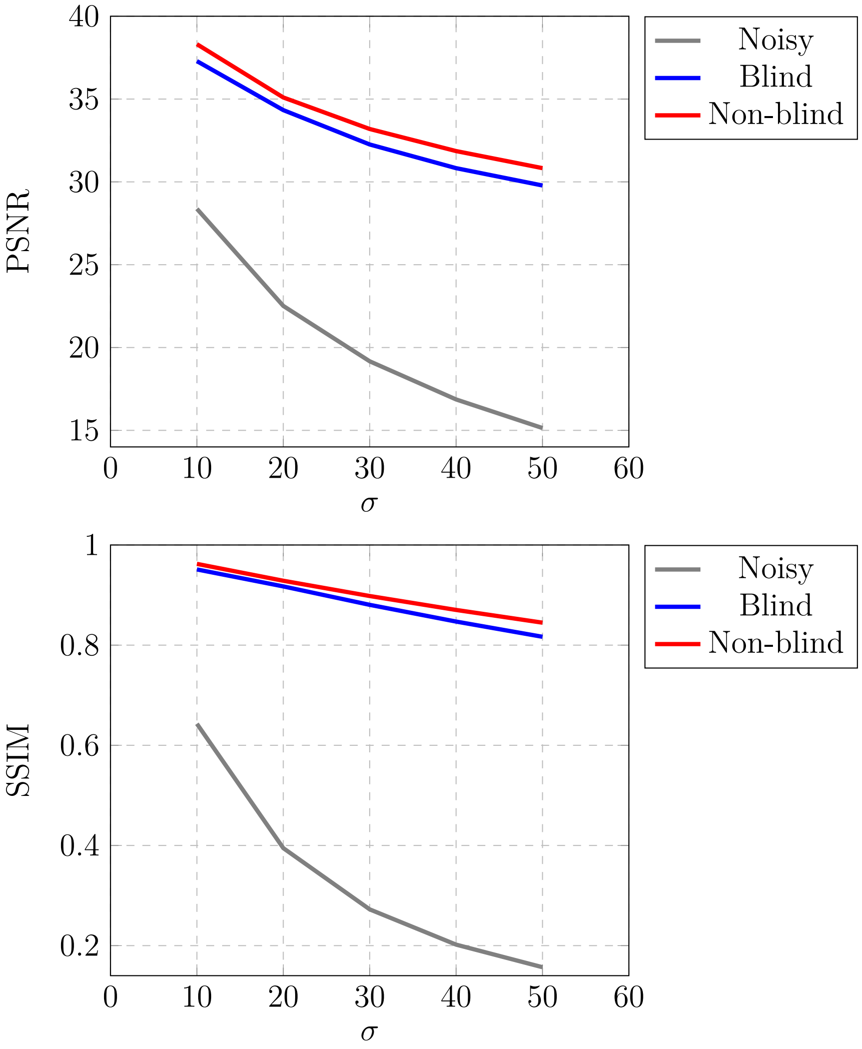


Figure 5.1: Performance achieved by FastDVDNet [28] in removing additive white Gaussian noise from videos as the value of the standard deviation σ changes. The non-blind model makes use of the noise map, while the blind model does not.

remove it because it is unable to estimate its intensity. On the contrary, the non-blind model is aware about the noise intensity, hence it can effectively remove more noise than the blind approach.

The distance between the lines representing the restoration performance of the two models in the PSNR plot is almost constant for all the values of σ, meaning that the ability of the network in removing noise decreases in the same way in both the approaches as the noise intensity increases. Instead, in the SSIM plot, the gap between the two lines increases as the value of σ grows up, indicating that the results produced by the blind model are more perceptually different than the ones produced by the non-blind model when the noise is severe.

Since the curve representing the noisy frames is much below the others, the network is able to remove the added noise. This claim is also confirmed by some qualitative

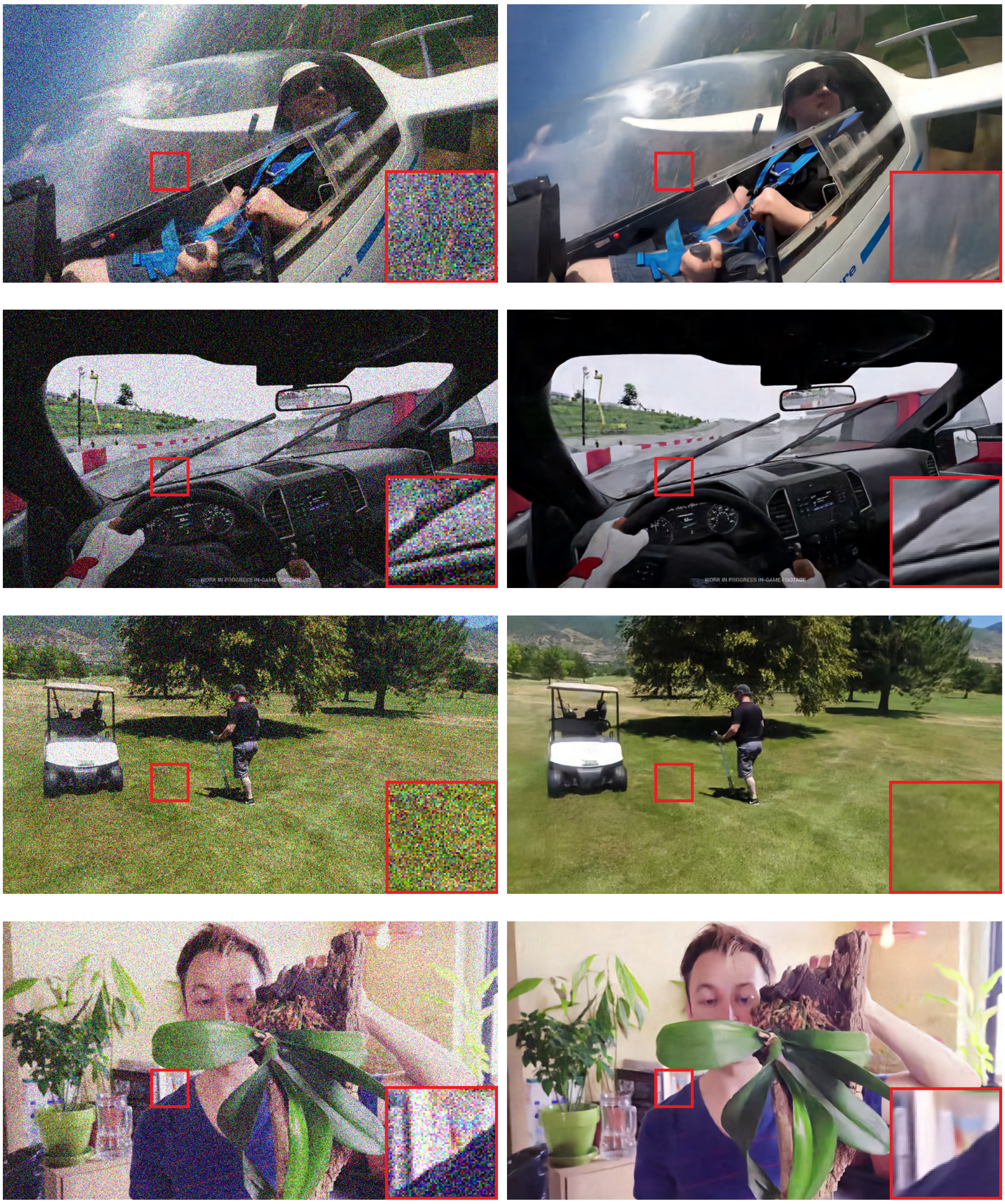


Figure 5.2: Qualitative results of FastDVDNet [28] in removing additive white Gaussian noise with $\sigma = 50$. Noisy frames on the left, restored frames on the right.

results shown in Figure 5.2.

For these reasons, it is possible to conclude that providing FastDVDNet [28] with information about the distortion intensity (the value of σ in this case) affecting the video sequence helps the network improve the denoising performance. It is important to notice that, although the training settings used in this experiment are very different from the ones used by Tassano et al. [28] in their paper (and reported in Subsection 4.3.6), the obtained denoising performance is very close to theirs in terms of PSNR. This means that the network is able to learn how to remove noise from videos even with much less training iterations (64000 instead of 320000).

5.4.2 Deblurring

The second experiment carried out aims to measure the capability of FastDVDNet [28] in removing out-of-focus blur from videos.

To simulate out-of-focus blur, the video sequences have been blurred using Gaussian kernels of different sizes and values of σ. More in detail, given a training sample consisting of five adjacent frames, each frame has been convolved with a Gaussian kernel whose value of σ is randomly extracted from the $[1, 6]$ range: $\sigma = 1$ means soft blur, $\sigma = 6$ means heavy blur. As for the denoising evaluation described in Subsection 5.4.1, all the frames within a training sample have been blurred using the same blur kernel, hence they all have the same blur intensity. Also in this case, other than evaluating the applicability of FastDVDNet [28] to videos affected by out-of-focus blur, the deblurring performance has been studied in the cases of blind and non-blind restoration.

Quantitative results are reported in Figure 5.5. The first observation is about the impact that using the noise map has on the deblurring performance of FastDVDNet [28]. In fact, the gap between the lines representing the restoration performance of the non-blind and blind models is quite high, meaning that providing the network with information about the blur affecting video sequences is highly beneficial and helps improve the deblurring performance. This is true both in terms of PSNR and SSIM [12]. Concerning PSNR, the non-blind restoration outperforms the blind one by about 2 dB for each value of σ tested. Regarding SSIM [12], the same consideration applies, as the non-blind restoration always outperforms the blind one by about 0.02. In addition, the gap is almost constant even though the value of σ increases, definitively demonstrating the superiority of the non-blind model with respect to the blind one.

Figure 5.4 shows qualitative results of blurred frames restored by FastDVDNet [28]. As shown, the network is able to recover the high-frequencies lost during the blurring process even if the blur introduced by Gaussian kernels is quite high ($\sigma = 5$ in this case).

In order to evaluate the effectiveness of FastDVDNet [28] in removing out-of-focus blur from videos, it is necessary to have at least a comparison method. However, in the literature, there is no recent work focusing on removing Gaussian blur from videos, as they all consider other types of blur (e.g., both EDVR [24] and DeBlurNet [17] restore videos affected by motion blur). Moreover, the recent state-of-the-art approaches for single image deblurring also focus on other types of blur kernels, different from the Gaussian one. For these reasons, the deblurring ability of FastDVDNet [28] has been compared with traditional non-blind single image deblurring methods, that is, the Wiener filtering, the Lucy-Richardson algorithm [63][64] and the Regularized filtering. All these methods perform non-

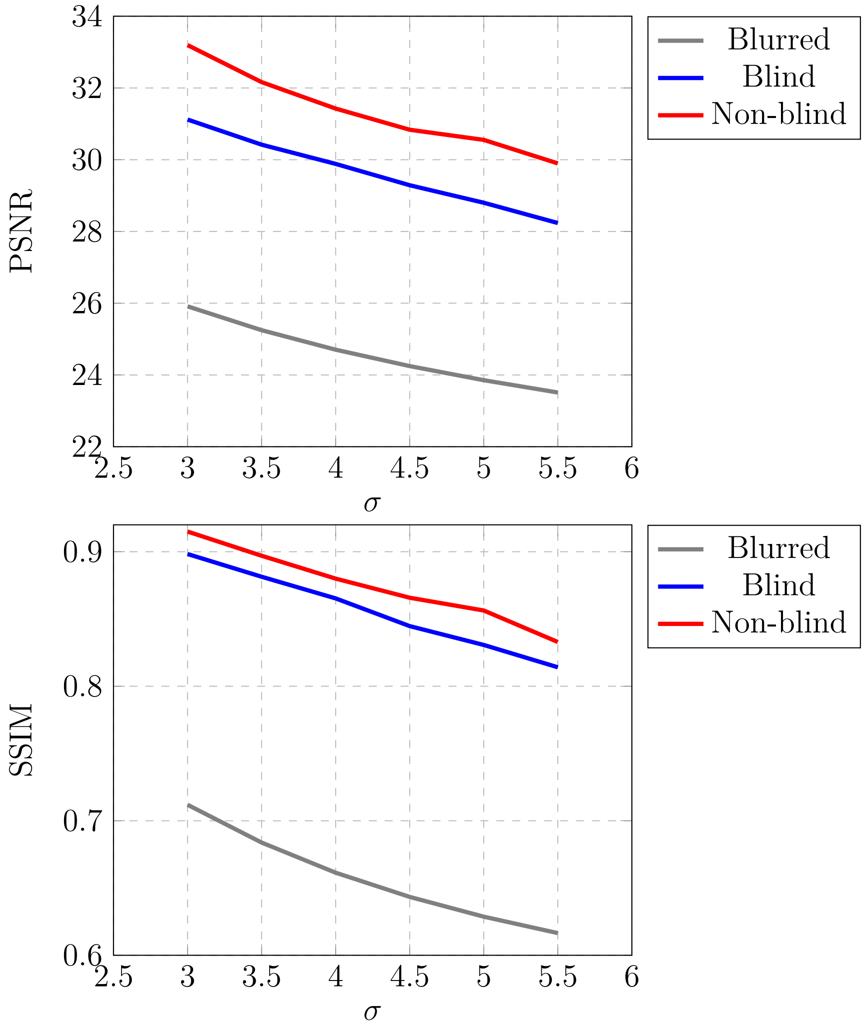


Figure 5.3: Performance achieved by FastDVDNet [28] in removing Gaussian blur from videos as the value of the standard deviation σ changes. The non-blind model makes use of the noise map, while the blind model does not.

blind deconvolutions and require the exact or approximated point-spread function (PSF) to recover the clean image from the blurred one.

One may notice that FastDVDNet [28] works on video sequences, while the other methods work on single images. Using sequences of five frames to carry out this evaluation may lead to an unfair comparison, as the network can exploit temporal redundancies while the other methods cannot, thus achieving worse deblurring performance. To address this problem, as FastDVDNet [28] is designed to take five consecutive frames as input, the simplest solution is that of using the same frame replicated five times so that it is not possible for the network to look for similar pixels in the other frames and exploit temporal redundancies. Therefore, for this experiment, the same frames replicated five times have been fed to the network to emulate single image deblurring.

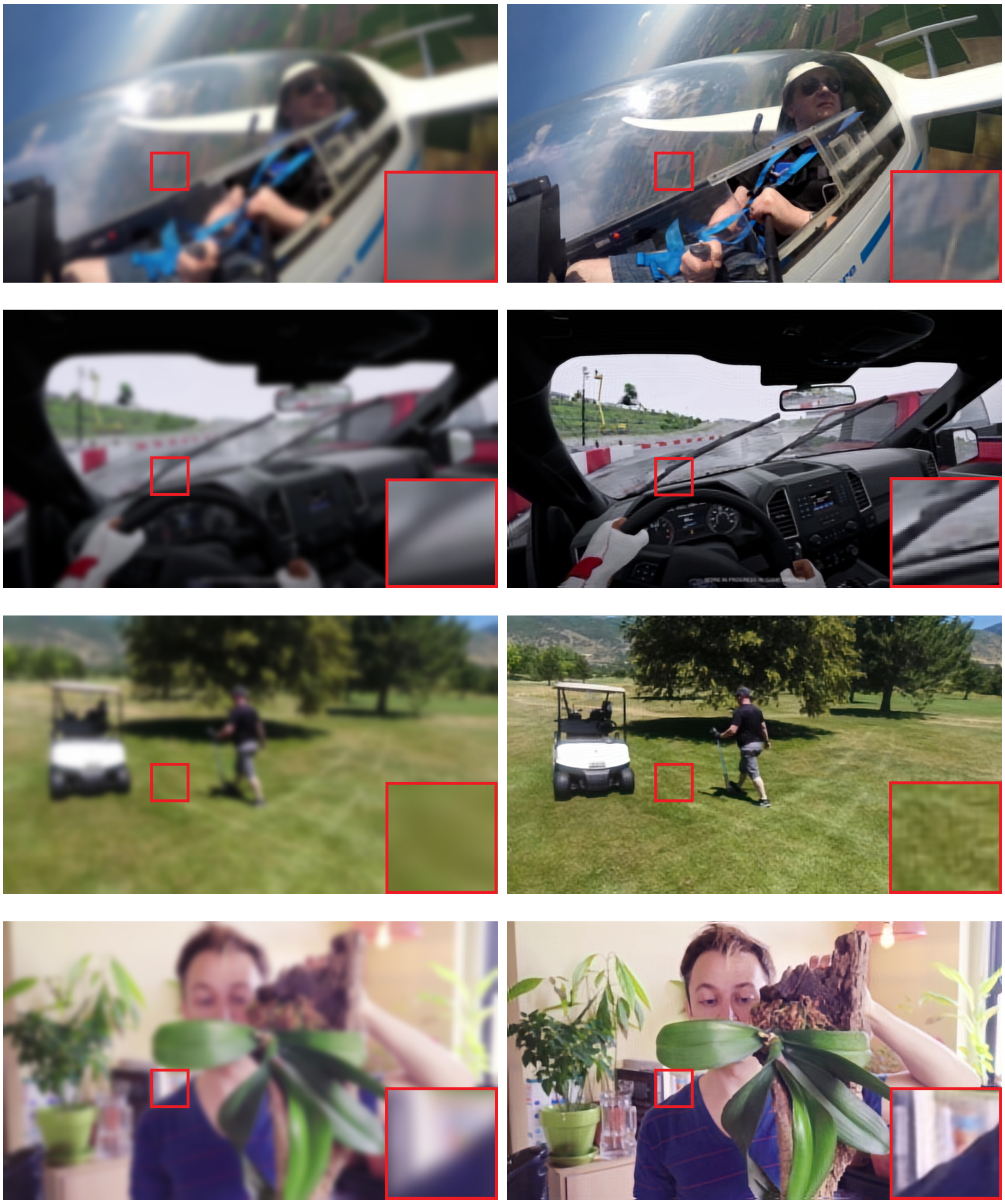


Figure 5.4: Qualitative results of FastDVDNet [28] in removing Gaussian blur with $\sigma = 5$. Blurred frames on the left, restored frames on the right.

The obtained results are reported in Table 5.1. Wiener filtering is the most effective method in removing Gaussian blur from images, both in terms of PSNR and SSIM [12]. However, it requires the exact PSF used to degrade the image, and it produces unacceptable results when another PSF is provided. The Lucy-Richardson algorithm [63][64] shows poor performance and it is the worst deblurring method among the compared ones. Besides, the Regularized filtering performance is very close to the deblurring performance obtained by FastDVDNet [28]. Concerning

Table 5.1: Comparison between FastDVDNet [28] and existing traditional deconvolution methods in restoring images affected by out-of-focus blur as the value of the standard deviation σ changes. The blind model makes use of the noise map, while the blind model does not.

Metric	σ	Weiner filtering	Lucy-Richardson algorithm [63][64]	Regularized filtering	**Blind FastDVDNet [28]**	**Non-blind FastDVDNet [28]**
PSNR	3	39.921	27.360	30.933	31.720	33.700
	3.5	38.503	26.553	30.282	30.919	32.898
	4	35.723	25.903	29.098	30.083	32.025
	4.5	34.924	25.354	28.744	29.388	31.299
	5	34.011	24.883	28.452	28.600	30.565
	5.5	33.871	24.472	28.198	28.035	29.817
SSIM	3	0.955	0.826	0.891	0.912	0.926
	3.5	0.952	0.807	0.880	0.900	0.914
	4	0.935	0.792	0.858	0.876	0.893
	4.5	0.929	0.779	0.850	0.864	0.877
	5	0.925	0.768	0.844	0.845	0.858
	5.5	0.921	0.758	0.838	0.826	0.839

PSNR, both the blind and the non-blind FastDVDNet [28] models outperform the Regularized filtering. In terms of SSIM [12], the latter is better than the blind FastDVDNet [28] model when the blur is severe (5.5 in this case).

Although in some cases traditional deconvolution methods are able to achieve better performance, they all require the PSF used to blur the images, in contrast to FastDVDNet [28] that has shown good performance even in the case of blind deblurring. Therefore, it is possible to conclude that FastDVDNet [28] is also able to effectively restore videos affected by out-of-focus blur with different level of severity in the case of non-blind restoration, but its deblurring performance considerably decreases when performing blind video deblurring. It is worth pointing out that the non-blind FastDVDNet [28] model requires just the value of σ instead of the whole degradation kernel.

5.4.3 Deblocking

The third experiment has the purpose of studying the performance of FastDVDNet [28] in restoring videos affected by compression artifacts.

Given a training sample consisting of five adjacent frames, each frame has been independently compressed using the JPEG compression algorithm with a random value of q extracted from the $[15, 35]$ range. Such lossy compression introduces visible blocking artifacts, whose size is 8×8. In contrast to the σ parameter of additive white Gaussian noise and Gaussian blur, here high values of q mean less compression, so the blocking artifacts become less visible and the deblocking performance is expected to increase.

The deblocking performance of FastDVDNet [28] is reported in Figure 5.5. Even

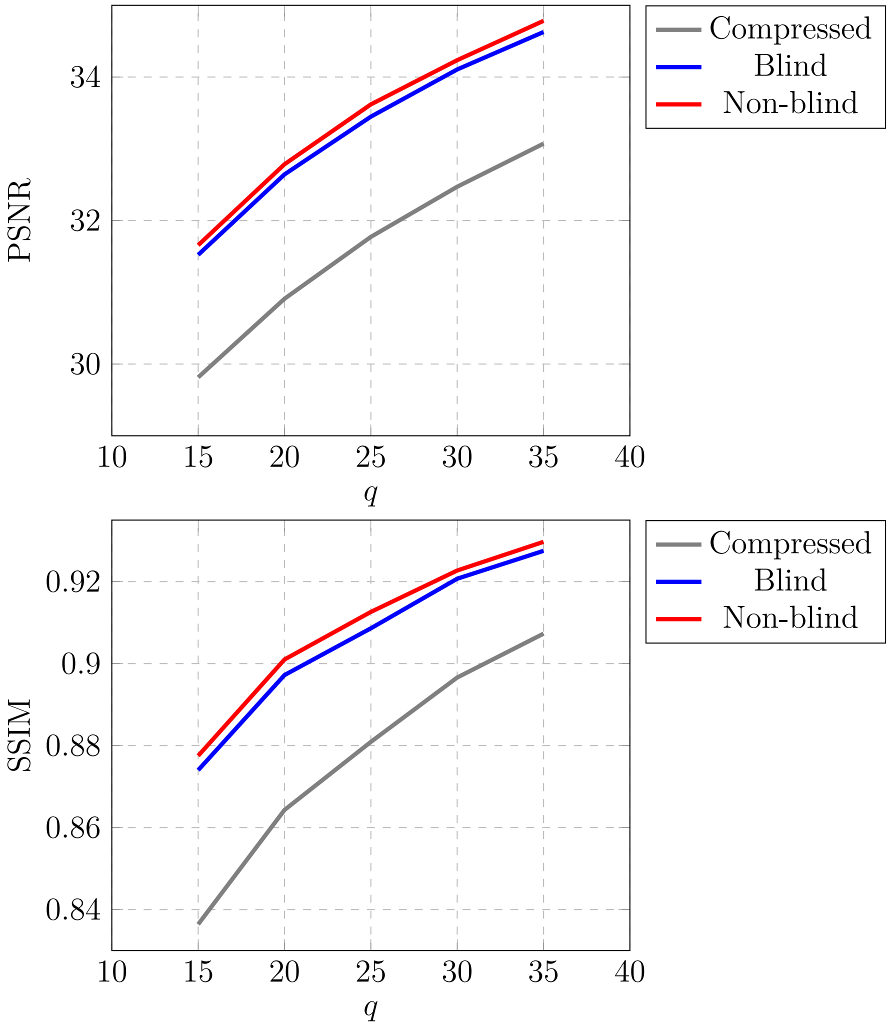


Figure 5.5: Performance achieved by FastDVDNet [28] in removing JPEG compression artifacts from videos as the value of the quality factor q changes. The non-blind model makes use of the noise map, while the blind model does not.

in this case, as happens for video denoising and deblurring, the non-blind model outperforms the blind one by about 0.1 dB in PSNR and about 0.003 in SSIM [12]. Despite this difference is quite low, using the noise map to provide the network with the q value used to compress frames of video sequences contributes to increase the deblocking performance.

Figure 5.6 shows four frames compressed with the JPEG algorithm and restored by FastDVDNet [28]. As shown, the network is able to remove the introduced blocking artifacts.

In order to understand the effectiveness of FastDVDNet [28] in removing JPEG compression artifacts from videos, it is necessary to compare it with some state-of-the-art approaches for compression artifact removal. In the literature, such methods mainly focus on other types of compression algorithms, such as AVC and

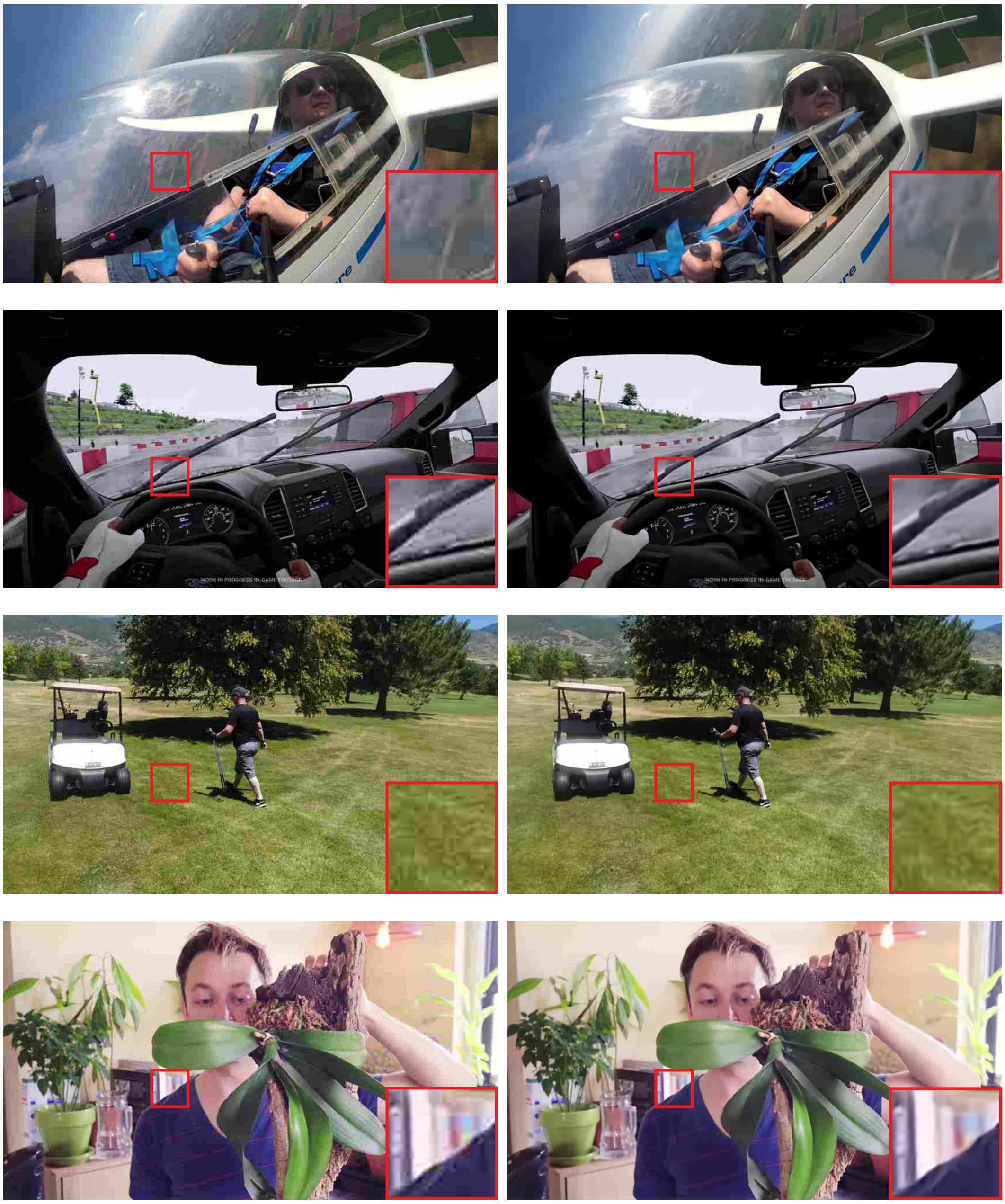


Figure 5.6: Qualitative results of FastDVDNet [28] in removing JPEG artifacts with $q = 15$. Compressed frames on the left, restored frames on the right.

HEVC, which exploit temporal redundancies among adjacent frames. Nevertheless, there are many works in the literature focusing on removing JPEG artifacts from images. For this reason, the deblocking performance of FastDVDNet [28] has been compared with the performance obtained by six state-of-the-art methods working on single images. These approaches have already been studied by Zini et al. [65] on the BSDS500 dataset [66] with specific values of the quality factor q. Evaluating FastDVDNet [28] on such dataset using those values of the quality factor allows a fair comparison. As for the comparison described in Subsection 5.4.2, FastDVDNet [28] has been evaluated on single images following the same procedure, that is, using the same frames replicated five times.

Table 5.2 reports the deblocking performance of FastDVDNet [28], ARCNN [4], DMCNN [67], MWCNN [68], S-NET [69], ARGAN [70] and RRDB [65] on the BSDS500 test set [66] using quality factors q equal to 10, 20 and 40. The reported performance refers to the Y channel of the YCbCr color space. It is worth noting that, except for FastDVDNet [28] and RRDB [65] that can handle different values of q with a single model, the values of PSNR and SSIM [12] in Table 5.2 for the other methods come from different models trained to deal with images compressed using a specific value of q. This means that, for example, a model of ARCNN [4] is trained on images compressed with $q = 10$ and another model is trained on images compressed with $q = 20$.

Table 5.2: Comparison between FastDVDNet [28] and existing deep learning methods in removing JPEG artifacts from the images of the BSDS500 test set [66]. The results refer to the Y channel of the YCbCr color space. The blind model makes use of the noise map, while the blind model does not.

Metric	q	ARCNN [4]	DMCNN [67]	MWCNN [68]	S-NET [69]	ARGAN [70]	RRDB [65]	**Blind FastDVDNet [28]**	**Non-blind FastDVDNet [28]**
PSNR	10	29.10	29.67	29.50	29.82	29.05	29.92	29.60	29.53
	20	31.25	31.98	31.34	32.15	31.23	32.23	32.02	32.03
	40	33.55	-	33.23	34.45	33.45	34.61	34.32	34.13
SSIM	10	0.819	0.840	0.835	0.844	0.806	0.847	0.834	0.834
	20	0.885	0.904	0.889	0.905	0.877	0.906	0.899	0.900
	40	0.929	-	0.928	0.941	0.923	0.943	0.937	0.933

The FastDVDNet [28] model has been trained using frames compressed with q within the $[15, 35]$ interval and it has never seen images compressed with q equals 10 or 40. Nevertheless, it manages to obtain state-of-the-art performance. For example, the blind FastDVDNet [28] model always outperforms ARCNN [4], MWCNN [68] and ARGAN [70] in terms of both PSNR and SSIM [12]. An interesting observation can be done about the performance obtained by the blind and non-blind FastDVDNet [28] models. In fact, the blind model outperforms the non-blind one when they are tested using q values never seen at training time. Filling the noise map with values of q never seen during training may confuse the model at inference time, thus causing a drop in its deblocking performance.

From these experiments it is possible to conclude that FastDVDNet [28] is able to effectively remove blocking artifacts caused by the JPEG algorithm from video sequences. Using the noise map to provide the network with the true value of the quality factor q used to compress the frames of a video sequence allows to increase the deblocking performance. The comparison with some deblocking methods working on single images has shown that FastDVDNet [28] can be applied also to images achieving state-of-the-art performance, outperforming some competitive approaches with a single model also when images are compressed using quality factors never seen during training. However, filling the noise map with values of q never seen during the training process causes a drop in the deblocking performance.

5.5 FastDVDNet on multiple artifacts

The experimental results reported in Section 5.4 have shown the flexibility of FastDVDNet [28] in removing different artifacts from videos and the superiority of the non-blind approach with respect to the blind one. The next step is that of assessing the ability of FastDVDNet [28] to restore videos affected by multiple artifacts. This because, if the network achieves very low performance when dealing with multiple distortions simultaneously, it could be difficult to use it as baseline model because there may be too many architectural modifications to make.

To this end, FastDVDNet [28] has been trained and evaluated on videos corrupted by multiple distortions, that is, noise followed by compression artifacts and blur followed by compression artifacts. The reason why only these two distortion combinations have been considered is explained in Section 2.4.

Note that the noise map, which is a single channel feature map containing information about distortion parameters, has been expanded from one channel to two channels, since there are two different artifacts affecting the video sequences at the same time. When performing blind restoration, both the channels are filled with zeros, whereas in the case of non-blind restoration the first channel is filled with the standard deviation of additive white Gaussian noise or Gaussian filters used to emulate out-of-focus blur and the second channel with the quality factor used by the JPEG compression algorithm.

5.5.1 Denoising and deblocking

The first experiment on multi-distorted videos aims to measure the effectiveness of FastDVDNet [28] in restoring noisy videos compressed using the JPEG algorithm. The training samples to carry out this evaluation have been generated first by adding additive white Gaussian noise, whose value of σ is randomly drawn from the $[5, 55]$ range, to the original training samples consisting of five adjacent frames, and then compressing each single noisy frame using the JPEG algorithm, with a random quality factor q in the $[15, 35]$ range. Also here, every frame within a training sample contains the same artifact intensity, that is, the σ parameter and the q factor used to introduce the artifacts are the same for all the frames of the training sample.

The restoration performance obtained by FastDVDNet [28] are reported in Table 5.3. Also in the case of multi-distorted videos, providing the network with the information about the distortion severity allows to improve the restoration performance. As expected, the best performance are obtained when the noise intensity is low and the quality factor is high (top-right cell of the table). The same consideration is also valid for the opposite case (bottom-left cell of the table). It can be observed that the performance increases moving along the anti-diagonal of the table, that is, as the σ value related to the noise intensity decreases and the q value related to the JPEG compression increases. Moreover, it is possible to see that the performance related to σ equal to 10 and 20 is very close to the performance reported in Figure

Table 5.3: Performance obtained by FastDVDNet [28] in restoring videos simultaneously affected by additive white Gaussian noise (with standard deviation σ) and JPEG compression artifacts (with quality factor q). The results are reported as PSNR/SSIM [12]. Each cell is composed of two rows: the first one represents the non-blind model, while the second one represents the blind model. The blind model makes use of the noise map, while the blind model does not.

σ \ q	15	20	25	30	35
10	31.894/0.882 31.167/0.858	32.937/0.901 32.348/0.881	33.647/0.912 33.047/0.895	34.161/0.920 33.559/0.903	34.551/0.924 33.901/0.908
20	31.608/0.874 30.922/0.847	32.391/0.888 31.940/0.876	32.879/0.896 32.447/0.886	33.208/0.901 32.799/0.892	33.448/0.905 33.007/0.894
30	31.068/0.858 30.567/0.842	31.641/0.869 31.135/0.850	31.969/0.874 31.505/0.857	32.177/0.878 31.755/0.863	32.333/0.881 31.894/0.867
40	30.465/0.839 29.918/0.817	30.876/0.847 30.268/0.823	31.103/0.852 30.588/0.832	31.242/0.854 30.813/0.839	31.337/0.856 30.919/0.842
50	29.842/0.820 29.373/0.802	30.154/0.826 29.669/0.807	30.306/0.829 29.910/0.815	30.396/0.831 30.061/0.820	30.461/0.832 30.129/0.822

5.5, indicating that the network can effectively remove JPEG artifacts even in the presence of mild noise.

In order to assess the effectiveness of FastDVDNet [28] in restoring noisy and compressed videos, a comparison with the state of the art is needed. However, a thorough research in the literature did not yield any result about approaches designed to perform multi-distorted video restoration considering noisy and compressed videos, except EDVR [24] that deals with other distortion types, i.e. motion blur and compression artifacts.

Since there are no methods for the comparison, the restoration performance of FastDVDNet [28] on multi-distorted videos has been compared with the performance obtained by placing two state-of-the-art approaches for video compression artifact reduction and video denoising in cascade. In this case, the output of the first model is used as input to the second one. Note that the first model should remove compression artifacts, while the second model should remove noise. This because, during the degradation process, noise is introduced before compression artifacts, therefore, during the restoration process, it is reasonable to remove compression artifacts first. In ideal conditions, one expects the first model to remove compression artifacts from the input frames, leaving the artifacts related to noise unchanged. Then, since the frames should contain just noise, the second model is expected to remove that noise, producing a clean video sequence.

Regarding the video compression artifact reduction method, the recent approaches in the literature do not address JPEG artifacts. However, FastDVDNet [28] trained to remove JPEG artifacts has shown its effectiveness in this task, allowing to achieve state-of-the-art performance. For this reason, the same non-blind model of FastDVDNet [28] employed for the evaluation in Subsection 5.4.3 has been

used. Concerning the video denoising method, the choice has directly fallen on FastDVDNet [28], as video denoising is its original task. In particular, the same non-blind model adopted for the evaluation in Subsection 5.4.1 as been used.
The results obtained by applying the aforementioned cascade of two artifact-specific models are reported in Table 5.4. At a glance, it is possible to notice that the performance obtained by the cascade of the two artifact-specific models is lower than the one obtained by using a single model to handle both the artifacts. This gap becomes more relevant when the intensity of the noise increases. In fact, the mean difference between the two approaches when σ is equal to 10 is about 0.7 dB for PSNR and 0.02 for SSIM [12], but it becomes about 3.8 dB and 0.21 when σ is equal to 50. This means that restoring videos simultaneously corrupted by multiple distortions using two artifact-specific methods in cascade is not an effective solution when the artifacts become strong.

Table 5.4: Comparison between non-blind FastDVDNet [28] and the two artifact-specific models placed in cascade in restoring videos simultaneously affected by additive white Gaussian noise (with standard deviation σ) and JPEG compression artifacts (with quality factor q). The results are reported as PSNR/SSIM [12]. Each cell is composed of two rows: the first one represents FastDVDNet [28], while the second one the cascade of the two artifact-specific models.

σ \ q	15	20	25	30	35
10	31.894/0.882 31.420/0.870	32.937/0.901 32.300/0.886	33.647/0.912 32.904/0.896	34.161/0.920 33.328/0.904	34.551/0.924 33.665/0.908
20	31.608/0.874 30.518/0.847	32.391/0.888 31.183/0.859	32.879/0.896 31.585/0.865	33.208/0.901 31.868/0.870	33.448/0.905 32.062/0.873
30	31.068/0.858 29.243/0.797	31.641/0.869 29.631/0.799	31.969/0.874 29.880/0.803	32.177/0.878 30.008/0.805	32.333/0.881 30.060/0.807
40	30.465/0.839 27.702/0.707	30.876/0.847 27.947/0.709	31.103/0.852 28.026/0.711	31.242/0.854 28.097/0.713	31.337/0.856 28.173/0.716
50	29.842/0.820 26.211/0.612	30.154/0.826 26.335/0.614	30.306/0.829 26.399/0.616	30.396/0.831 26.450/0.617	30.461/0.832 26.512/0.619

An example of a frame restored by the cascade of the two artifact-specific models is shown in Figure 5.7. Here, only patches of size 64 × 64 extracted from the full frames are reported to better see the results. There are four patches in the figure: the first image represents the distorted frame, affected by noise and compression artifacts; the second image represents the output of the first model, which aims to remove compression artifacts; the third image, corresponding to the restored frame, represents the output of the second model, which aims to remove noise; the fourth image corresponds to the ground truth. As shown, the first model, which expected a frame corrupted just by JPEG artifacts, failed its task because of the presence of excessive noise. As a consequence, the second model, which expected a frame corrupted only by noise, was not able to remove the remaining artifacts, leaving the frame with visible distortions.

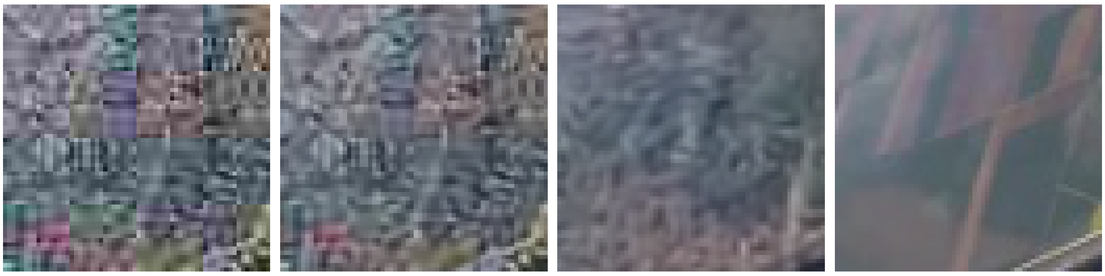


Figure 5.7: Example of a patch affected by additive white Gaussian noise ($\sigma = 50$) and JPEG compression artifacts ($q = 15$) restored by a cascade of two artifact-specific models. The first model removes compression artifacts, while the second model removes noise. The first image represents the distorted patch, the second image represents the output of the first model, the third image represents the output of the second model, corresponding to the restored patch, while the fourth image represents the ground truth.

From these experimental results it is possible to conclude that FastDVDNet [28] is able to restore multi-distorted videos when they are corrupted by noise and compression artifacts. Even in this case, as happens for single artifacts, providing the network with information about the distortion intensity allows to increase the restoration performance. Finally, trying to restore videos affected by noise and compression artifacts using a cascade of two artifact-specific models, i.e. a model for video compression artifact reduction and a model for video denoising, is not an effective solution especially in the presence of strong artifacts, suggesting that using a single model to handle both the distortions is the best solution.

5.5.2 Deblurring and deblocking

The second experiment on multi-distorted videos aims to assess whether FastDVDNet [28] is able to restore videos simultaneously affected by blur and compression artifacts. Although the results are promising in the case of noise and compression artifacts, it may happen that the network is unable to restore videos affected by other distortion combinations. Indeed, working on blurred videos requires the network to recover high-frequency components, which is a difficult task, and adding compression artifacts makes this task even tougher.

The training samples for this experiment have been generated by blurring every frame of the original training samples using Gaussian kernels, whose values of σ are randomly extracted from the $[1, 6]$ range. The obtained blurred frames are then compressed by applying the JPEG algorithm using a random quality factor q in the $[15, 35]$ range. Even in this case, each frame of a training sample contains the same distortion intensity.

The restoration performance obtained using FastDVDNet [28] properly trained to handle Gaussian blur and JPEG related artifacts is reported in Table 5.5. Also in this case, two models performing non-blind and blind restoration have been compared. As for all the experiments carried out so far, the non-blind model outperforms the blind one. However, the restoration performance on this specific combination of distortions is much lower than the performance obtained on videos corrupted by noise and compression artifacts, reported in Subsection 5.5.1. Indeed, the best values of PSNR and SSIM [12] are 28.368 dB and 0.774, which are even lower than the worst values of PSNR and SSIM [12] reported in Table 5.3 related to noisy and compressed videos, which are 29.373 dB and 0.802, respectively. Especially for SSIM [12], which is a perceptive measure, 0.774 means that the difference between the output frames and the ground truth is highly perceived by the human eye. An example of a blurred and compressed frame restored by FastDVDNet [28] is shown in Figure 5.8. As shown, the network is able to almost completely remove the JPEG artifacts, but it fails to recover high-frequency components leaving the frame blurred.

Table 5.5: Performance obtained by FastDVDNet [28] in restoring videos simultaneously affected by Gaussian blur (with standard deviation σ) and JPEG compression artifacts (with quality factor q). The results are reported as PSNR/SSIM [12]. Each cell is composed of two rows: the first one represents the non-blind model, while the second one represents the blind model. The blind model makes use of the noise map, while the blind model does not.

σ \ q	15	20	25	30	35
3	27.106/0.729 26.715/0.717	27.592/0.745 27.178/0.735	27.931/0.756 27.498/0.745	28.178/0.766 27.731/0.756	28.368/0.774 27.895/0.763
3.5	26.597/0.708 26.224/0.698	27.045/0.724 26.626/0.715	27.348/0.734 26.904/0.723	27.579/0.744 27.110/0.733	27.744/0.751 27.264/0.739
4	26.158/0.691 25.827/0.682	26.568/0.706 26.178/0.698	26.847/0.714 26.427/0.705	27.043/0.723 26.596/0.714	27.744/0.751 26.726/0.719
4.5	25.765/0.676 25.506/0.668	26.144/0.690 25.829/0.683	26.393/0.697 26.043/0.690	26.585/0.705 26.204/0.697	26.709/0.711 26.316/0.702
5	25.404/0.661 25.207/0.655	25.757/0.675 25.550/0.670	25.991/0.681 25.761/0.676	26.163/0.690 25.904/0.684	26.276/0.694 26.006/0.688
5.5	25.065/0.648 24.835/0.640	25.401/0.662 25.227/0.656	25.628/0.667 25.455/0.662	25.785/0.675 25.610/0.669	25.883/0.679 25.710/0.674

The obtained results allow to conclude that, although FastDVDNet [28] is effective in restoring multi-distorted videos simultaneously corrupted by noise and compression artifacts, it is not when videos are affected by blur and compression artifacts. Besides, since the results using this combination of distortions are not promising even when the artifacts are mild, such artifact combination has not been further investigated in the following experiments.

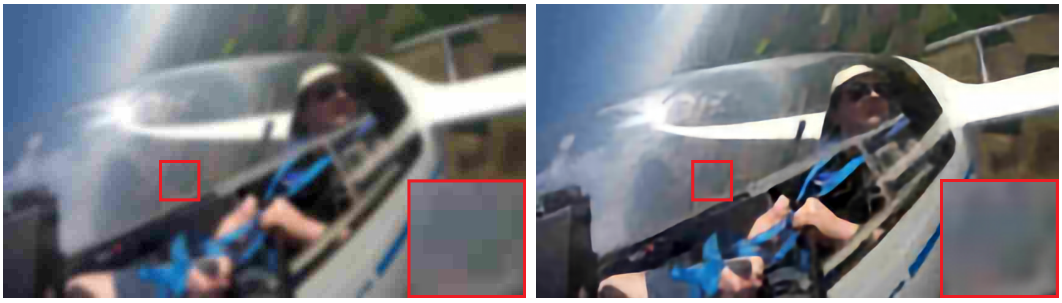


Figure 5.8: Example of a multi-distorted frame (left) affected by Gaussian blur ($\sigma = 4$) and JPEG compression artifacts ($q = 15$) restored by FastDVDNet [28] (right)

5.6 Distortion parameter estimation

The results reported in Section 5.4 concerning the effectiveness of FastDVDNet [28] in restoring videos affected by single distortions, i.e. noise, blur or compression artifacts, show the superiority of the non-blind approach over the blind one. In fact, the performance obtained when FastDVDNet [28] is provided with degradation operator information is better than the one obtained when no information is given to the network.

The noise map contains information about the degradation operator parameters used to degrade the video sequences, making the network aware about the intensity of the artifacts. However, while at training time such parameters are known, at inference time they may not and, thus, an external resource to estimate them is required. This problem has been discussed in Subsection 4.4.2, in which a CNN, dubbed DPEN, has been devised to estimate the different distortion parameters required by MdVRNet.

A set of experiments have been conducted in order to evaluate the accuracy of DPEN in predicting the intensity of the distortions affecting video sequences, both using single and multiple degradation operators, and its applicability in being integrated into existing blind frameworks as a source of additional information to increase restoration performance.

5.6.1 Parameter estimation on single distortions

The first experiment concerning DPEN aims to evaluate its accuracy in predicting the artifact intensity when videos are affected by single distortions. Given an input frame, DPEN is able to extract the parameters of specific distortions affecting it, such as the σ value of additive white Gaussian noise and the quality factor q used by the JPEG algorithm. Note that, when considering single-distorted videos, DPEN must be trained to deal with that specific distortion. For instance, if the considered task is video denoising, DPEN must be trained to predict the σ value of the additive white Gaussian noise affecting the video sequence. Therefore, to conduct these experiments, three different DPEN models have been trained according to the considered distortion type.

Regarding the training process of DPEN, all the models have been trained for 500 epochs on patches of size 64 × 64 randomly extracted from the DAVIS 2017 train set [61], using a learning rate initially set to $1e^{-4}$, L_1 as loss function and Adam [62] as optimizer. The learning rate has been reduced by a factor of 10 whenever the loss function did not decrease for 20 consecutive epochs. Concerning the distortion parameters, the same values described in Section 5.4 have been used to degrade the frames, that is, $\sigma \in [5, 55]$ for denoising, $\sigma \in [1, 6]$ for deblurring and $q \in [15, 35]$ for deblocking.

Table 5.6 shows the performance, in terms of mean absolute error (MAE), achieved by the DPEN models in predicting the distortion parameters of the three considered degradation operators. Since MAE measures the error between the prediction and the ground truth, the lower the value, the better. As shown, DPEN is able to infer quite accurate values of the σ parameter for both additive white Gaussian noise and Gaussian blur. More in detail, concerning the σ parameter of AWGN, the error increases as the noise intensity increases, while in the case of Gaussian blur the error is almost constant as the severity of the blur becomes stronger. Concerning the quality factor q used by the JPEG compression algorithm, DPEN infers values with an error of about 2. Moreover, the error increases as the value of q increases. This is due to the fact that, when the quality factor is quite high, blocking artifacts are not as pronounced as they are when the value is low.

Table 5.6: Mean absolute error (MAE) of the three DPEN models in estimating the distortion parameters of additive white Gaussian noise (AWGN), Gaussian blur and JPEG compression

AWGN		Gaussian blur		JPEG compression	
σ	MAE	σ	MAE	q	MAE
10	0.708	1.5	0.223	15	1.897
20	0.824	2.5	0.225	20	1.877
30	0.947	3.5	0.238	25	2.189
40	1.101	4.5	0.272	30	2.269
50	1.243	5.5	0.209	35	3.750

It is important to point out that DPEN is trained using patches of size 64 × 64. However, additional experimental results have shown that applying DPEN to images of different sizes, i.e. 480 × 854 in the case of the DAVIS 2017 dataset [61], considerably increases the error. To solve this problem, the adopted solution is that of decomposing the target image into 64 × 64 non-overlapping patches, estimating the distortion parameters on single patches using DPEN and finally averaging the obtained estimations. Such procedure allows to obtain an error as close to zero as possible, since there are some patches on which DPEN overestimates the distortion parameter and others on which DPEN underestimates it, thus computing the mean value allows to better approximate the real one.

5.6.2 FastDVDNet on single distortions using the estimated distortion parameters

The experimental results reported in Subsection 5.6.1 have shown that DPEN is able to predict quite accurate values of different distortion parameters on frames corrupted by single distortions. Further experiments have been carried out to understand whether DPEN can be integrated into existing blind restoration frameworks to provide them with additional information that could increase their restoration performance.

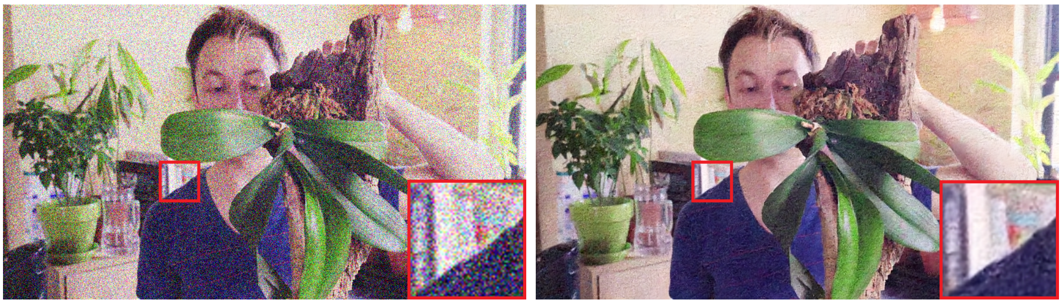


Figure 5.9: Example of a noisy frame (left) restored by FastDVDNet [28] (right) using the true distortion parameter value at training time and the parameter inferred by DPEN at test time

Before starting with these experiments, it is important to comprehend whether DPEN should be used only at inference time or also at training time. This because it is possible that using different information for the same distortion at inference time may confuse the network and cause a drop in the restoration performance. DPEN is able to predict distortion parameter values up to a certain precision limit. Although such error may appear quite low, the network may fail the restoration process because, during training, it has learned a mapping between a value, corresponding to the distortion parameter, and the intensity of artifacts in the video sequence and, at inference time, such mapping is broken. This statement is exemplified in Figure 5.9, which shows a noisy frame restored by FastDVDNet [28] trained using the true distortion parameters and tested using the parameters inferred by DPEN. As shown, the network is not able to completely remove the noise from the frame. This is due to the fact that, at training time, there is no error concerning the distortion parameter used, while at inference time the error introduced by DPEN, never seen during training, breaks the learned mapping. This consideration suggests that using the value inferred by DPEN to fill the noise map also at training time should preserve the mapping learned between the estimated distortion parameter and the intensity of the artifacts in the video sequences and, hence, the restoration performance is expected to increase. According to this consideration, FastDVDNet [28] has been trained using the distortion parameters predicted by DPEN.
The restoration performance obtained by FastDVDNet [28] using the distortion

Table 5.7: Comparison among the performance obtained by three different FastDVDNet [28] models, i.e. the non-blind model, the blind model and the model using the distortion parameters inferred by DPEN, on different restoration tasks involving single-distorted video sequences. The non-blind model uses the true distortion parameter values, while the blind model does not use any additional information. The results are reported as PSNR/SSIM [12].

Denoising

σ	True information (non-blind)	No information (blind)	Information estimated by DPEN
10	38.315/0.962	37.291/0.951	35.384/0.909
20	35.096/0.929	34.333/0.917	34.180/0.919
30	33.193/0.898	32.259/0.881	32.224/0.885
40	31.860/0.870	30.829/0.847	30.836/0.851
50	30.830/0.845	29.782/0.817	29.741/0.818

Deblurring

σ	True information (non-blind)	No information (blind)	Information estimated by DPEN
3	33.196/0.915	31.120/0.898	31.117/0.900
3.5	32.163/0.897	30.420/0.881	30.469/0.864
4	31.426/0.880	29.884/0.865	29.761/0.837
4.5	30.836/0.866	29.289/0.845	28.941/0.810
5	30.553/0.856	28.801/0.831	28.251/0.784
5.5	29.897/0.833	28.236/0.814	27.267/0.745

Compression artifact reduction

q	True information (non-blind)	No information (blind)	Information estimated by DPEN
15	31.656/0.878	31.522/0.874	31.273/0.870
20	32.784/0.901	32.643/0.897	32.014/0.894
25	33.619/0.913	33.448/0.909	33.012/0.903
30	34.235/0.923	34.106/0.921	33.791/0.916
35	34.784/0.930	34.628/0.928	34.119/0.920

parameters estimated by DPEN on different restoration tasks is reported in Table 5.7. To better show how the performance changes when the network is provided with the true information, the information estimated by DPEN and no information about the distortion parameters related to the artifacts affecting the video sequences, also the results already presented in Section 5.4 are reported in the table.

As expected, the best restoration performance is obtained by providing the network with the true values of the distortion parameters (non-blind approach). However, this result does not have much value in this case because, as mentioned, having exact information about the artifacts affecting a video sequence is not always possible. Interestingly, using the parameter estimation made by DPEN leads to a decline in performance. Indeed, in all the three restoration tasks, even the performance obtained by the blind approach is higher than the one obtained using the information inferred by DPEN. The most plausible reason to explain this fact is related to the errors made by DPEN in predicting the distortion parameter values. Although these errors are quite low, as discussed in Subsection 5.6.1, the negative impact they have is higher than expected. This means that wrong information confuses the network preventing the correct restoration process, and it is better to provide no information at all instead of providing distorted one.

5.6.3 Parameter estimation on multiple distortions

When video sequences are corrupted by single degradation operators, the introduced artifacts are different from the ones introduced by multiple degradation operators, both in shape and distribution. This means that, although DPEN has shown accurate results in predicting the degradation operator parameters on videos affected just by single distortions, the prediction accuracy may be very different when multiple artifacts are present.

In order to evaluate the effectiveness of DPEN in predicting distortion information in the case of multi-distorted videos, two distinct models have been trained and evaluated considering the noise/compression and the blur/compression combinations. To carry out these experiments, DPEN has been modified to output two parameters, corresponding to σ and q, where σ is the parameter related to either additive white Gaussian noise or Gaussian blur depending on the considered artifact combination.

The training process is equal to the one described in Subsection 5.6.1, but with a difference about the loss function used. Since there are two parameters to estimate, two L_1 loss functions are computed between the regressed values and their corresponding real values, and the final loss is simply the summation of these two loss functions.

The performance measured in MAE obtained by DPEN in predicting the distortion parameters of two different distortion combinations, i.e. noise/compression and blur/compression, is reported in Table 5.8. Concerning the combination noise/compression, it is possible to notice that the error made in estimating the σ value is much higher than the error made when the frame is corrupted just by noise, as reported in Table 5.6. Indeed, the maximum error increased from 1.24 to 4.57. Interestingly, while in the previous case the error made increases as the degradation intensity increases, here the opposite happens, i.e. the stronger the noise level, the lower the error made by DPEN. In addition, the error decreases as the quality factor q increases. The estimated q related to JPEG artifacts is quite precise, especially in the presence of strong noise, and the MAE is very similar to the MAE reported in Table 5.6. This means that DPEN is not sensitive to noise when inferring the distortion parameter related to compression artifacts. Furthermore, as happens for single distortions, the higher the compression, the higher the error made in predicting the q parameter.

Regarding the blur/compression combination, the error made in estimating the σ parameter of Gaussian blur increases by about 0.2 with respect to the error

Table 5.8: Mean absolute error of the two DPEN models in estimating the distortion parameters on videos affected by two distortion combinations, i.e. noise/compression and blur/compression

Noise/compression combination

Estimated parameter: σ (AWGN)

σ \ q	15	20	25	30	35
10	4.567	4.124	3.754	3.498	3.469
20	4.131	3.613	3.285	3.099	3.118
30	3.738	3.367	3.057	2.899	2.903
40	3.566	3.281	3.103	2.780	2.882
50	3.993	2.888	2.697	2.452	2.464

Estimated parameter: q (JPEG)

σ \ q	15	20	25	30	35
10	3.023	1.835	2.179	2.671	4.280
20	1.795	1.568	1.937	2.065	3.007
30	1.409	1.254	1.644	1.772	2.313
40	1.267	1.254	1.623	1.604	2.139
50	1.322	1.164	1.629	1.546	1.711

Blur/compression combination

Estimated parameter: σ (Gaussian blur)

σ \ q	15	20	25	30	35
1.5	0.480	0.448	0.417	0.407	0.395
2.5	0.496	0.463	0.415	0.404	0.401
3.5	0.470	0.426	0.395	0.377	0.362
4.5	0.461	0.416	0.398	0.394	0.381
5.5	0.628	0.531	0.517	0.480	0.461

Estimated parameter: q (JPEG)

σ \ q	15	20	25	30	35
1.5	2.212	2.340	2.164	2.179	3.688
2.5	1.832	1.890	1.913	1.877	2.722
3.5	1.589	1.553	1.817	1.745	2.348
4.5	1.471	1.453	1.746	1.695	2.251
5.5	1.493	1.462	1.801	1.725	2.146

related to blurred and uncompressed frames. Also in this case, the error decreases when the value of q increases. Besides, the q parameter is more accurate when the compression is high, as also happens for the noise/compression combination.

Even in the presence of multiple artifacts, at inference time, using DPEN on frames with a size greater than 64 × 64 leads to less accurate results. Therefore, the same procedure described in Subsection 5.6.1 should be adopted, that is, the tested frame is decomposed into 64 × 64 non-overlapping patches, each of which is processed by DPEN generating two values and the final values are obtained by averaging all the values related to the first distortion and all the values related to the second one.

5.6.4 FastDVDNet on multiple distortions using the estimated distortion parameters

Although using the information extracted by DPEN has led to worse restoration performance in the case of single distortions, as demonstrated by the experiments in Subsection 5.6.2, it may happen that providing the network with approximated information is better than using no information at all when videos are affected by multiple distortions, because of the additional complexity of the artifacts introduced by multiple degradation operators. To verity this claim, FastDVDNet [28] has been trained on video sequences affected by multiple artifacts using DPEN as distortion parameter estimator.

The restoration performance obtained on videos simultaneously corrupted by additive white Gaussian noise and JPEG artifacts is reported in Table 5.9, which shows how the performance changes as the artifacts become stronger. Note that the first two rows of each cell are the same as the ones in Table 5.3, but they are also reported here to better see the difference of the three approaches. The first consideration is related to the non-blind approach that, even in this case, shows the best performance. Interestingly, in contrast to the results obtained in Subsection 5.6.2 concerning single-distorted videos, estimating the distortion parameters affecting the video sequences on multi-distorted videos allows to increase the restoration capability of the network. Indeed, the performance obtained by FastDVDNet [28] using DPEN as source of information about degradation operators is always higher than the one of the blind approach in terms of PSNR, while the SSIM [12] is lower in only a few cases. This means that, when dealing with multi-distorted videos, providing the network with approximated information about the degradation operators is better than providing no information at all. This may be related to the fact that combinations of degradation operators introduce more complex artifacts than single operators, and using approximated information

Table 5.9: Comparison among the performance obtained by three different FastDVDNet [28] models, i.e. the non-blind model, the blind model and the model using the distortion parameters inferred by DPEN, on video sequences simultaneously affected by additive white Gaussian noise (with standard deviation σ) and JPEG compression artifacts (with quality factor q). The non-blind model uses the true distortion parameter values, while the blind model does not use any additional information. The results are reported as PSNR/SSIM [12]. Each cell is composed of three rows: the first one represents the non-blind model, the second one represents the blind model, while the third one represents the model using the distortion parameters estimated using DPEN.

σ \ q	15	20	25	30	35
10	31.894/0.882 31.167/0.858 31.678/0.876	32.937/0.901 32.348/0.881 32.574/0.889	33.647/0.912 33.047/0.895 33.100/0.899	34.161/0.920 33.559/0.903 33.735/0.909	34.551/0.924 33.901/0.908 34.221/0.915
20	31.608/0.874 30.922/0.847 31.106/0.863	32.391/0.888 31.940/0.876 32.040/0.878	32.879/0.896 32.447/0.886 32.549/0.886	33.208/0.901 32.799/0.892 32.877/0.890	33.448/0.905 33.007/0.894 33.094/0.893
30	31.068/0.858 30.567/0.842 30.710/0.848	31.641/0.869 31.135/0.850 31.296/0.857	31.969/0.874 31.505/0.857 31.643/0.863	32.177/0.878 31.755/0.863 31.878/0.867	32.333/0.881 31.894/0.867 32.005/0.869
40	30.465/0.839 29.918/0.817 30.098/0.827	30.876/0.847 30.268/0.823 30.582/0.837	31.103/0.852 30.588/0.832 30.826/0.841	31.242/0.854 30.813/0.839 30.982/0.845	31.337/0.856 30.919/0.842 31.041/0.845
50	29.842/0.820 29.373/0.802 29.468/0.808	30.154/0.826 29.669/0.807 29.899/0.817	30.306/0.829 29.910/0.815 30.047/0.820	30.396/0.831 30.061/0.820 30.159/0.822	30.461/0.832 30.129/0.822 30.208/0.823

helps the network understand how to remove them.
The results reported in Subsection 5.5.2 have shown that FastDVDNet [28] is not effective in restoring videos affected by blur and compression artifacts, even when the network receives the exact information about the distortion parameters σ and q. In all the experiments conducted so far, non-blind models have shown better performance with respect to the others. This because the network is provided with the real information about the degradation operators affecting the video sequences and, hence, it learns how to exploit such information to properly remove the artifacts. Using the distortion parameters estimated by DPEN would lead to restoration performance lower than the one obtained by the non-blind approach, which is already quite low, because the network would be provided with noisy information. For this reason, FastDVDNet [28] has not been trained and evaluated using the distortion parameters inferred by DPEN for the blur/compression combination.
The experiments carried out using DPEN to provide FastDVDNet [28] with information about degradation operator parameters, when handling multi-distorted videos, have shown a performance improvement with respect to the blind approach, meaning that this strategy could be adopted by blind methods to increase the effectiveness in restoring videos corrupted by multiple distortions.

5.7 MdVRNet on multiple artifacts

So far, several experiments have been conducted in order to verify both the flexibility of FastDVDNet [28] in performing restoration tasks different from the one it is designed for, i.e. video denoising, and its applicability in restoring multi-distorted videos. The network has shown high flexibility, as it outperforms traditional methods in removing out-of-focus blur and deep learning approaches in removing JPEG compression artifacts. Concerning multi-distorted video restoration, the obtained results demonstrated that using a network to remove multiple artifacts simultaneously is better than using two artifact-specific models in cascade, also revealing the effectiveness of FastDVDNet [28] in handling multiple distortions. In addition, the experiments described in Section 5.6 demonstrated that estimating the degradation operator parameters using DPEN and providing them to FastDVDNet [28] allows to increase its restoration performance when videos are corrupted by noise and compression artifacts.
The aforementioned considerations are fundamental for the proposed MdVRNet framework. Indeed, one of the main ideas behind this network is that of incorporating a preliminary step aiming to estimate the degradation operator parameters to make the network aware about the distortion intensity and help it improve the quality of the results. Moreover, since MdVRNet inherits the main characteristics of FastDVDNet [28], the effectiveness of the baseline model in removing multiple artifacts from videos is inherited, too, providing a lower bound to the restoration performance.
The main improvements over the FastDVDNet [28] architecture are related to the introduction of an original distortion parameter estimation module, i.e. DPEN, and

the novel multi-scale restoration block, which extends the original denoising block of FastDVDNet [28] by adding a full-resolution stream, for detail preservation, and an attention mechanism to weight the features coming from the two parallel branches according to their importance in reconstructing the target frame. More detailed information can be found in Section 4.4.

Table 5.10: Comparison between MdVRNet and FastDVDNet [28] using the distortion parameters estimated by DPEN in restoring videos simultaneously affected by additive white Gaussian noise (with standard deviation σ) and JPEG compression artifact (with quality factor q). The results are reported as PSNR/SSIM [12]. Each cell is composed of two rows: the first one represents MdVRNet, while the second one represents FastDVDNet [28]

σ \ q	15	20	25	30	35
10	31.861/0.881 31.678/0.876	32.886/0.900 32.574/0.889	33.592/0.912 33.100/0.899	34.111/0.920 33.735/0.909	34.484/0.924 34.221/0.915
20	31.600/0.874 31.106/0.863	32.479/0.889 32.040/0.878	32.984/0.897 32.549/0.886	33.306/0.902 32.877/0.890	33.561/0.906 33.094/0.893
30	31.083/0.857 30.710/0.848	31.725/0.869 31.296/0.857	32.049/0.874 31.643/0.863	32.266/0.878 31.878/0.867	32.431/0.881 32.005/0.869
40	30.495/0.838 30.098/0.827	30.931/0.847 30.582/0.837	31.154/0.851 30.826/0.841	31.309/0.854 30.982/0.845	31.425/0.857 31.041/0.845
50	29.776/0.816 29.468/0.808	30.140/0.823 29.899/0.817	30.288/0.826 30.047/0.820	30.408/0.829 30.159/0.822	30.501/0.831 30.208/0.823

In order to evaluate how the proposed improvements impact the restoration performance, new experiments have been conducted. In particular, the contribution of the preliminary distortion parameter estimation step has been assessed by comparing MdVRNet with the model trained without any additional information about degradation operators, that is, the blind model using the noise map filled with zeros. Besides, the contribution of the multi-scale restoration block has been measured by comparing MdVRNet with FastDVDNet [28] using the information extracted by the same DPEN model, thus preventing the difference in performance to be attributed to errors in predicting the distortion parameters.

Regarding the training process, both MdVRNet and FastDVDNet [28] have been trained using the training settings described in Section 5.3 and the same training samples generated by adding additive white Gaussian noise with σ randomly extracted from the $[5, 55]$ range to the clean frames and compressing the resulting frames using the JPEG algorithm with a random q value in the $[15, 35]$ range.

The performance obtained by MdVRNet in restoring videos corrupted by noise and compression artifacts is reported in Table 5.10. As shown by the results in the table, MdVRNet outperforms FastDVDNet [28] both when the artifacts are mild and severe. More in detail, the multi-scale restoration blocks in MdVRNet allow to improve the FastDVDNet [28] performance by about 0.35 dB and 0.01 in terms of PSNR and SSIM [12], respectively. This improvement is quite constant

for all the values of σ and q controlling the artifact intensity. This means that the novel multi-scale restoration block allows MdVRNet to obtain better videos than the ones obtained using FastDVDNet [28], both perceptually and in terms of reconstruction error, regardless of the degradation severity.
The results of the investigation about the contribution that estimating the distortion parameter has on the performance are reported in Table 5.11. As shown, using DPEN to provide MdVRNet with information about the distortion intensity improves the performance. Indeed, PSNR improves by about 0.2 dB and SSIM [12] by about 0.01 on average. More in detail, the improvement in PSNR is higher when the artifacts are stronger, since in this case the average improvement is about 0.3 dB. This means that the use of DPEN is particularly useful to reduce reconstruction errors when the distortions are severe. Concerning SSIM [12], the performance improvement is constant for all the parameter values tested.

Table 5.11: Comparison between two MdVRNet models, i.e. the model using the distortion parameters inferred by DPEN and the blind model, in restoring videos simultaneously affected by additive white Gaussian noise (with standard deviation σ) and JPEG compression artifacts (with quality factor q). The results are reported as PSNR/SSIM [12]. Each cell is composed of two rows: the first one represents MdVRNet using the information provided by DPEN, while the second one represents the blind MdVRNet model.

σ q	15	20	25	30	35
10	31.861/0.881 31.776/0.874	32.886/0.900 32.830/0.893	33.592/0.912 33.527/0.904	34.111/0.920 34.014/0.913	34.484/0.924 34.375/0.916
20	31.600/0.874 31.444/0.868	32.479/0.889 32.266/0.880	32.984/0.897 32.732/0.887	33.306/0.902 33.042/0.893	33.561/0.906 33.257/0.896
30	31.083/0.857 30.883/0.852	31.725/0.869 31.457/0.860	32.049/0.874 31.784/0.866	32.266/0.878 31.984/0.870	32.431/0.881 32.107/0.873
40	30.495/0.838 30.277/0.831	30.931/0.847 30.652/0.839	31.154/0.851 30.864/0.844	31.309/0.854 30.998/0.847	31.425/0.857 31.075/0.849
50	29.776/0.816 29.564/0.810	30.140/0.823 29.868/0.817	30.288/0.826 30.006/0.820	30.408/0.829 30.100/0.822	30.501/0.831 30.162/0.824

Qualitative results of MdVRNet are illustrated in Figure 5.10, which shows four different video frames corrupted by noise and compression artifacts (left) and restored by the network (right). It is possible to see that most of the artifacts have been removed from the distorted frames. However, there are still some problems related to detail reconstruction. This issue is better exemplified in Figure 5.11, which shows 64×64 patches extracted from the frames in Figure 5.10. As shown, the finer details are not properly reconstructed during the restoration process. For instance, the fine details in the third example, representing a lawn of grass, are confused with noise and they have not been properly restored. It is worth pointing out that the artifacts affecting the frames in the figure are very strong ($\sigma = 50$ and $q = 15$ in this case) and the examples have just the purpose of showing how the network

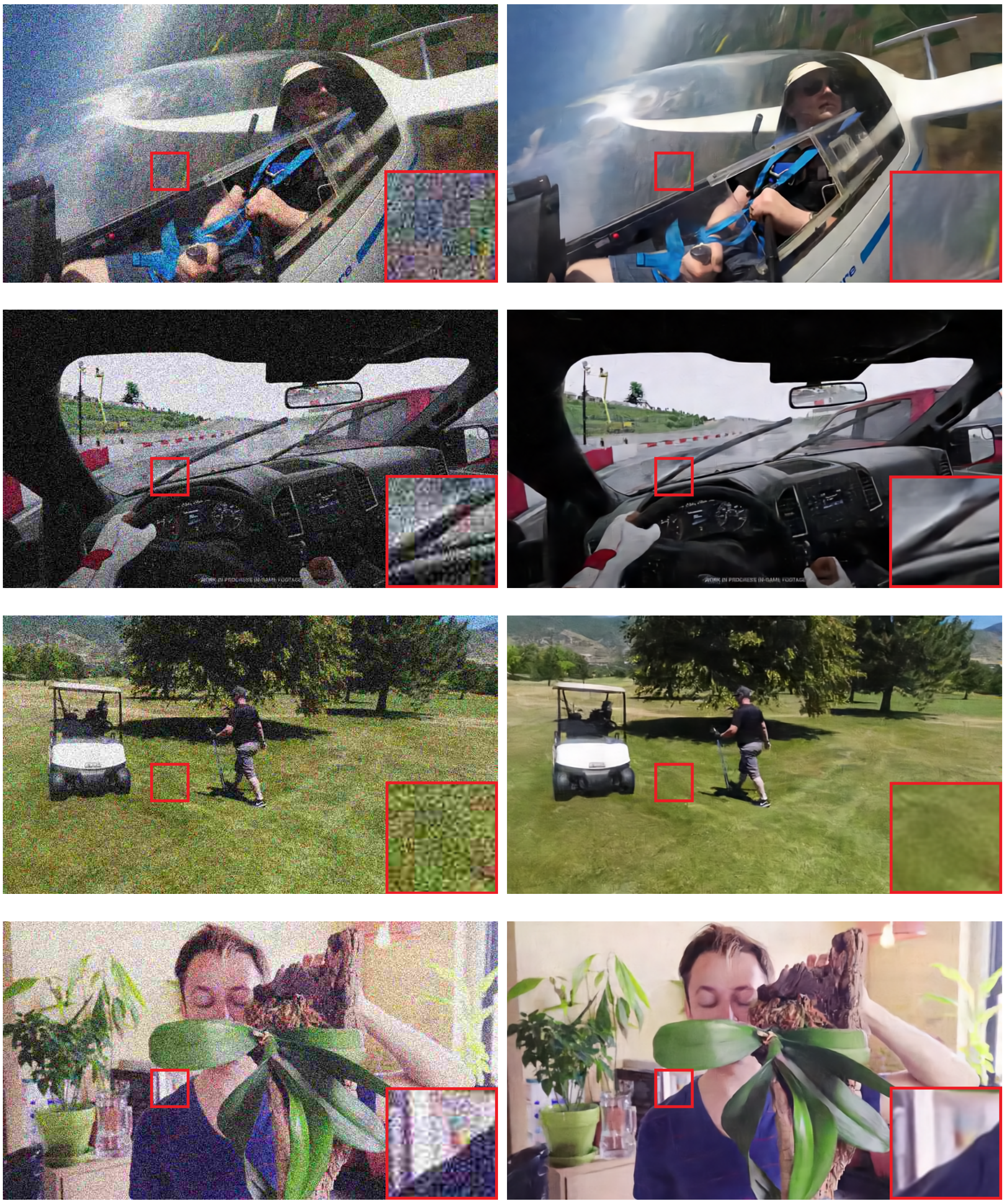


Figure 5.10: Qualitative results of MdVRNet in restoring videos simultaneously affected by additive white Gaussian noise ($\sigma = 50$) and JPEG compression artifacts ($q = 15$). Distorted frames on the left, restored frames on the right.

behaves in extreme cases but, as reported in Table 5.10, decreasing the intensity of the artifacts allows to obtain better results, both in terms of reconstruction error and perceptually.

The computational time required by the proposed MdVRNet to restore video frames of different resolutions on a Tesla P100-PCIE-16GB GPU is reported in Table 5.12. As shown, although MdVRNet outperforms FastDVDNet [28] in terms

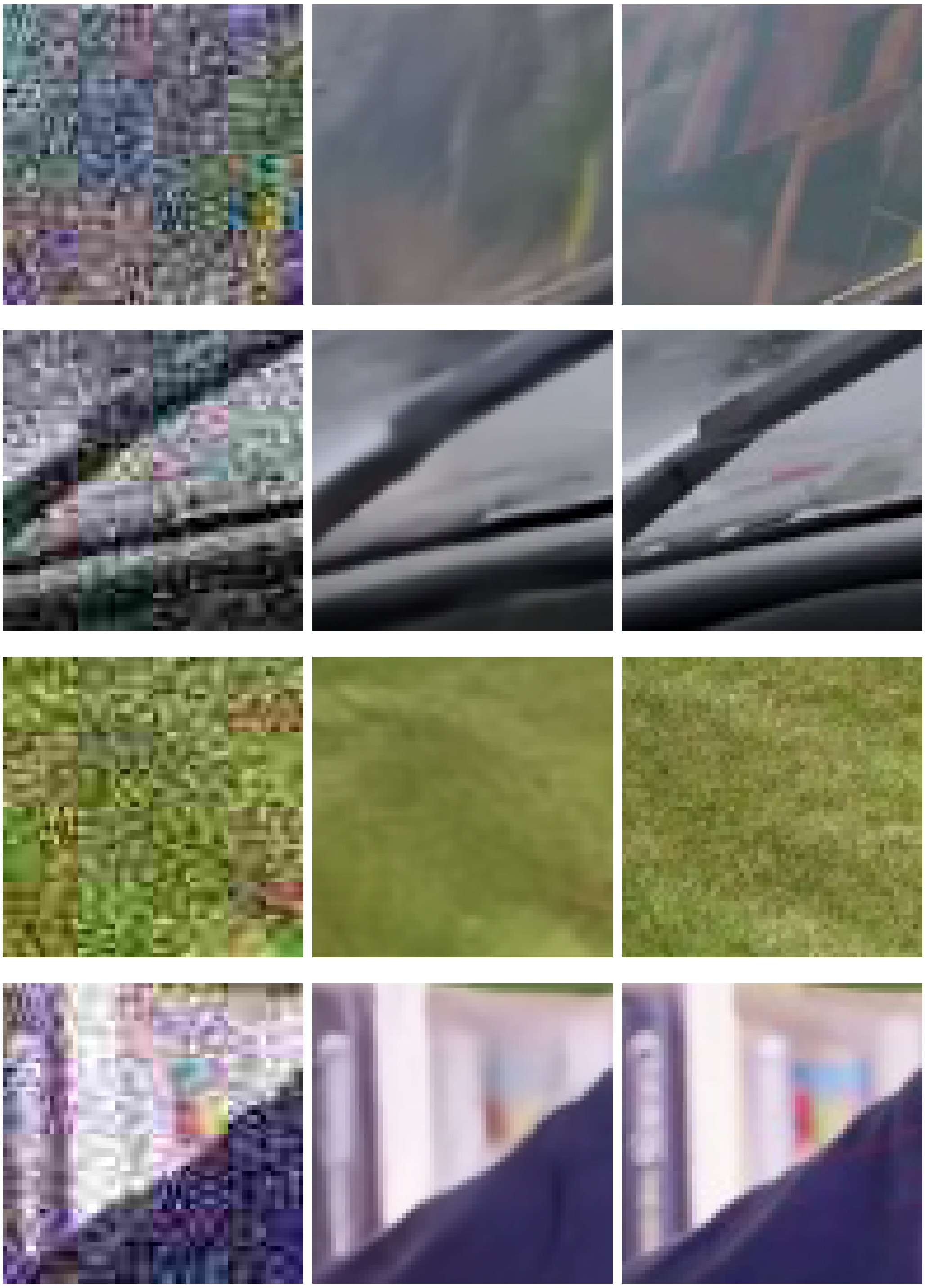


Figure 5.11: Example of patches affected by additive white Gaussian noise ($\sigma = 50$) and JPEG compression artifacts ($q = 15$) restored by MdVRNet. Distorted frames on the left, restored frames on the center, ground truth on the right.

of effectiveness, it does not in terms of efficiency, as the latter is about three times faster. This increase in computational time is caused by the full-resolution branch of the multi-scale restoration blocks. Indeed, working at full-resolution considerably increases the number of operations to perform that, in turn, increases the computational time. In addition, also the distortion parameter estimation step performed using DPEN has a considerable impact on the overall computational time. In fact, about 30% of the total time is dedicated to estimate distortion parameters. The reason why such step takes so long is related to the strategy adopted to process frames for the estimation of the degradation parameters. As mentioned in Subsection 5.6.3, for more accurate estimations, the target frame is decomposed into 64 × 64 non-overlapping patches and each patch is processed by DPEN, considering the average values as the final values. However, the number of patches to be processed increases with frame resolution. Therefore, an alternative strategy that avoids such patch decomposition may almost completely remove this additional overhead.

Table 5.12: Comparison between the restoration time required by FastDVDNet [28] and MdVRNet in restoring multi-distorted videos. The results are reported in seconds. The computational time required by MdVRNet considers both the time required for the distortion parameter estimation and the time required for the actual restoration process.

Frame size	FastDVDNet [28]	MdVRNet (distortion parameter estimation + restoration)
256 × 448	$0.023s$ [43 FPS]	$0.066s$ ($0.016s + 0.050s$) [15 FPS]
480 × 854	$0.068s$ [14 FPS]	$0.227s$ ($0.073s + 0.154s$) [4 FPS]
720 × 1280	$0.135s$ [7 FPS]	$0.490s$ ($0.167s + 0.323s$) [2 FPS]

In conclusion, the experimental results demonstrated that MdVRNet outperforms FastDVDNet [28] in restoring videos affected by multiple distortions, noise and compression artifacts to be precise, thanks to the proposed improvements related to both the degradation parameter estimation module, i.e. DPEN, specifically devised for this task and the multi-scale restoration block allowing to extract features at different scales and properly fuse them according to the importance they have in reconstructing the target frame. However, MdVRNet is less efficient than FastDVDNet [28], as the overall restoration process is about three times slower. This is due to both the full-resolution branch within the multi-scale restoration blocks and to the strategy adopted for the distortion parameter estimation using DPEN.

Chapter 6

Conclusions

This thesis addressed the problem of video restoration using convolutional neural networks.

In the literature, several methods to deal with different video restoration tasks have been proposed, and the most recent ones are based on deep learning due to its incredible success in many computer vision tasks. However, although they are able to produce high-quality results, they have been designed to restore videos affected just by single distortions and they cannot be used to restore multi-distorted videos, i.e. videos corrupted by multiple degradation operators.

To address this limitation, this thesis proposes a new deep neural network, called Multi-distorted Video Restoration Network and abbreviated MdVRNet, which exploits some of the best components of state-of-the-art video restoration approaches in order to effectively restore videos affected by multiple distortions.

The most promising methods for video restoration in the literature have been critically analyzed under different aspects to better understand how they work, the key ideas behind them and their basic components. This analysis allowed to understand how the studied methods perform important operations, such as pixel motion estimation and frame alignment, and to extract the building blocks so that they can be assembled to build a new video restoration framework. It emerged that all the video restoration methods can be divided into two classes based on how they perform motion estimation and frame alignment, which can be done explicitly using a specific module or implicitly by the network itself. In addition, another possible distinction is between non-blind methods and blind methods: the former exploit information about degradation operators, while the latter do not. Both the classes have their own advantages and disadvantages.

Among the studied methods, a promising approach for video denoising, i.e. Fast-DVDNet [28], has been selected as baseline model for further analysis. The best characteristics of this network have been exploited for the design of MdVRNet.

MdVRNet is a two-stage restoration network that progressively aligns adjacent frames, allowing to extract both spatial and temporal information from the target frame and its adjacent ones. The first restoration stage pays more attention to single pixel restoration, due to its limited temporal information, while the second restoration stage pays more attention to restoring local areas, as it has a more complete vision of the scene. To make the restoration process more robust, an original distortion parameter estimation module, called Distortion Parameter Estimation Network and dubbed DPEN, has been devised and integrated within the framework itself to provide it with information about the intensity of the artifacts affecting the video sequence. In addition, a novel multi-scale restoration block allowing to extract features at different scales using two parallel streams has been designed, so that each stream can extract different but complementary features. More in detail, the full-resolution stream learns fine pixel dependencies for finer detail reconstruction, while the low-resolution stream learns coarse pixel dependencies to make the most of the semantic in local areas. The features of these two streams are then weighted according to their importance in reconstructing the target frame using an attention mechanism and fused to obtain a degradation map, which is finally subtracted from the degraded target frame to restore it.

Several experiments have been carried out with different purposes. First of all, the flexibility of the baseline model in restoring videos affected by single distortions, i.e. noise, blur and compression artifacts, has been assessed by comparing the network with both traditional and deep learning approaches. As a result, FastDVDNet [28] outperformed existing traditional methods in removing out-of-focus blur and also existing deep learning methods in removing compression artifacts, demonstrating its flexibility in being adapted to other tasks different from the one it is designed for. Then, the adaptability of the network to perform video restoration even when videos are corrupted by multiple degradation operators has been evaluated, considering the artifacts introduced by noise/compression and blur/compression. Since there is no method in the literature to deal with these artifacts simultaneously, the easiest solution to remove them is that of using two artifact-specific models sequentially, that is, the output of the first model is used as input to the second one. However, the obtained results demonstrated that it is not a good solution, as the restored frames still presented visible artifacts. Instead, FastDVDNet [28] properly trained to remove a specific combination of artifacts allows to obtain much better results, confirming that using a single model to jointly remove the artifacts is better than using a cascade of two artifact-specific models. Experimental results focusing on how the performance changes when the information about degradation intensity is estimated by DPEN have shown that, when dealing with single distortions, using no information is better than using approximated one. Conversely, when videos are corrupted by multiple distortions, using the parameters estimated by DPEN leads to an increase of the restoration performance, due to the additional complexity of the artifacts introduced by multiple degradation operators. Motivated by the promising results of FastDVDNet [28] in restoring multi-distorted videos and by the interesting results obtained using DPEN to estimate distortion parameters, MdVRNet has been trained and evaluated to remove noise and compression artifacts from videos. The obtained results have demonstrated that MdVRNet is more effective than FastDVDNet [28] in restoring multi-distorted videos. Further experiments have been carried out to better understand the contribution the additional distortion parameter estimation stage and the multi-scale restoration block have on the restoration performance, and the obtained results have shown that they contribute almost equally. However, although the proposed MdVRNet outperforms FastDVDNet [28] in terms of effectiveness, it does not in terms of efficiency, as it processes videos with a lower frame rate than the baseline model. Such difference is due to both the full-resolution branch within the multi-scale restoration blocks, which inevitably increases the computational time, and the strategy adopted to estimate distortion parameters, which decomposes the input frame into non-overlapping patches for more accurate estimations.

The main limitations of the proposed MdVRNet are the assumption of globally distributed artifacts and the computational time required to perform the restoration process, leading to several possible future developments. First, it would be interesting to make MdVRNet spatially varying so that it can restore videos even if the intensity of the artifacts is different in different spatial locations, i.e. noise is stronger in dark regions, because in real cases the artifacts may not be globally distributed within video frames. This indirectly means adapting the architecture of DPEN to make it able to predict punctual or local values instead of global values. Moreover, network compression methods may be used to reduce the network dimension without losing effectiveness, and further studies may be conducted to understand how to increase the prediction accuracy of DPEN without decomposing the target frame into patches, allowing to decrease the computational time of the entire MdVRNet framework. Another possible future work may be that of investigating other video compression algorithms, such as HEVC. Such methods exploit temporal redundancies to increase the compression rate, but they introduce new types of artifacts on which MdVRNet has never been tested.

Bibliography

[1] Yuzhen Lu. “Out-of-focus Blur: Image De-blurring”. In: *ArXiv* abs/1710.00620 (2017).

[2] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. “Learning a Deep Convolutional Network for Image Super-Resolution”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 184–199. ISBN: 978-3-319-10593-2.

[3] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. “Image super-resolution via sparse representation”. In: *IEEE transactions on image processing* 19.11 (2010), pp. 2861–2873.

[4] Chao Dong, Yubin Deng, Chen Change Loy, and Xiaoou Tang. “Compression Artifacts Reduction by a Deep Convolutional Network”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 576–584. DOI: 10.1109/ICCV.2015.73.

[5] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising”. In: *IEEE Transactions on Image Processing* 26.7 (2017), pp. 3142–3155. ISSN: 1941-0042. DOI: 10.1109/tip.2017.2662206. URL: http://dx.doi.org/10.1109/TIP.2017.2662206.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[7] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.1556.

[8] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. ICML’15. Lille, France: JMLR.org, 2015, pp. 448–456.

[9] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. “Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network”. In: June 2016. DOI: `10.1109/CVPR.2016.207`.

[10] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. “Deep Generative Adversarial Compression Artifact Removal”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 4836–4845. DOI: `10.1109/ICCV.2017.517`.

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. “Generative Adversarial Networks”. In: *Advances in Neural Information Processing Systems* 3 (June 2014). DOI: `10.1145/3422622`.

[12] Zhou Wang, Alan Bovik, Hamid Sheikh, and Eero Simoncelli. “Image Quality Assessment: From Error Visibility to Structural Similarity”. In: *Image Processing, IEEE Transactions on* 13 (May 2004), pp. 600–612. DOI: `10.1109/TIP.2003.819861`.

[13] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. “Deep multi-scale convolutional neural network for dynamic scene deblurring”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 3883–3891.

[14] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William Freeman. “Video Enhancement with Task-Oriented Flow”. In: *International Journal of Computer Vision* 127 (Aug. 2019). DOI: `10.1007/s11263-018-01144-2`.

[15] Anurag Ranjan and Michael J Black. “Optical flow estimation using a spatial pyramid network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4161–4170.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. “Spatial Transformer Networks”. In: *Advances in Neural Information Processing Systems 28 (NIPS 2015)* (June 2015).

[17] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. “Deep Video Deblurring for Hand-Held Cameras”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 237–246. DOI: `10.1109/CVPR.2017.33`.

[18] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. “Real-Time Video Super-Resolution with Spatio-Temporal Networks and Motion Compensation”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2848–2857. DOI: 10.1109/CVPR.2017.304.

[19] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. “Deep Video Super-Resolution Network Using Dynamic Upsampling Filters Without Explicit Motion Compensation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3224–3232. DOI: 10.1109/CVPR.2018.00340.

[20] Bert Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. “Dynamic Filter Networks”. In: *Neural Information Processing Systems (NIPS)* (Jan. 2016).

[21] Matias Tassano, Julie Delon, and Thomas Veit. “DVDNET: A Fast Network for Deep Video Denoising”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. 2019, pp. 1805–1809. DOI: 10.1109/ICIP.2019.8803136.

[22] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. “DeepFlow: Large Displacement Optical Flow with Deep Matching”. In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 1385–1392. DOI: 10.1109/ICCV.2013.175.

[23] Michele Claus and Jan van Gemert. “ViDeNN: Deep Blind Video Denoising”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019, pp. 1843–1852. DOI: 10.1109/CVPRW.2019.00235.

[24] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. “Edvr: Video restoration with enhanced deformable convolutional networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019, pp. 1954–1963.

[25] *NTIRE19: New Trends in Image Restoration and Enhancement workshop and challenges on image and video restoration and enhancement in conjunction with CVPR 2019*. URL: https://data.vision.ee.ethz.ch/cvl/ntire19/ (visited on 04/07/2021).

[26] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. “Deformable Convolutional Networks”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 764–773. DOI: 10.1109/ICCV.2017.89.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. “Attention is All You Need”. In: NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.

[28] Matias Tassano, Julie Delon, and Thomas Veit. "Fastdvdnet: Towards real-time deep video denoising without flow estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020, pp. 1354–1363.

[29] Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. "Spatio-Temporal Deformable Convolution for Compressed Video Quality Enhancement". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (Apr. 2020), pp. 10696–10703. DOI: 10.1609/aaai.v34i07.6697.

[30] Zhenyu Guan, Qunliang Xing, Mai Xu, Ren Yang, Tie Liu, and Zulin Wang. "MFQE 2.0: A New Approach for Multi-frame Quality Enhancement on Compressed Video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (Oct. 2019), pp. 1–1. DOI: 10.1109/TPAMI.2019.2944806.

[31] Ren Yang, Mai Xu, Zulin Wang, and Tianyi Li. "Multi-frame quality enhancement for compressed video". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018, pp. 6664–6673.

[32] Sepp Hochreiter and Michael Mozer. "A Discrete Probabilistic Memory Model for Discovering Dependencies in Time". In: *Proceedings of the International Conference on Artificial Neural Networks.* ICANN '01. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 661–668. ISBN: 3540424865.

[33] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K. Cormack. "Study of Subjective and Objective Quality Assessment of Video". In: *IEEE Transactions on Image Processing* 19.6 (2010), pp. 1427–1441. DOI: 10.1109/TIP.2010.2042111.

[34] Sachin Mehta, Amit Kumar, Fitsum Reda, Varun Nasery, Vikram Mulukutla, Rakesh Ranjan, and Vikas Chandra. "EVRNet: Efficient Video Restoration on Edge Devices". In: *arXiv preprint arXiv:2012.02228* (2020).

[35] Andrea Polesel, Giovanni (Gianni) Ramponi, and V John Mathews. "Image Enhancement via Adaptive Unsharp Masking". In: *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 9 (Feb. 2000), pp. 505–10. DOI: 10.1109/83.826787.

[36] Edwin Land and John McCann. "Lightness and Retinex Theory". In: *Journal of the Optical Society of America* 61 (Feb. 1971), pp. 1–11. DOI: 10.1364/JOSA.61.000001.

[37] Li Tao, Chuang Zhu, Guoqing Xiang, Yuan Li, Huizhu Jia, and Xiaodong Xie. "LLCNN: A convolutional neural network for low-light image enhancement". In: *2017 IEEE Visual Communications and Image Processing (VCIP).* IEEE. 2017, pp. 1–4.

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 1–9.

[39] Feifan Lv, Feng Lu, Jianhua Wu, and Chongsoon Lim. "MBLLEN: Low-Light Image/Video Enhancement Using CNNs." In: *BMVC*. 2018, p. 220.

[40] Wenjing Wang, Chen Wei, Wenhan Yang, and Jiaying Liu. "GLADNet: Low-Light Enhancement Network with Global Awareness". In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 2018, pp. 751–755. DOI: `10.1109/FG.2018.00118`.

[41] Young Moon, Bok Han, Hyeon Yang, and Ho Lee. "Low Contrast Image Enhancement Using Convolutional Neural Network with Simple Reflection Model". In: *Advances in Science, Technology and Engineering Systems Journal* 4 (Feb. 2019). DOI: `10.25046/aj040115`.

[42] Chun-Wei Liu and Tyng-Luh Liu. "A sparse linear model for saliency-guided decolorization". In: *2013 IEEE International Conference on Image Processing*. 2013, pp. 1105–1109. DOI: `10.1109/ICIP.2013.6738228`.

[43] Kaiming He, Jian Sun, and Xiaoou Tang. "Guided Image Filtering". In: *IEEE transactions on pattern analysis and machine intelligence* 35 (June 2013), pp. 1397–1409. DOI: `10.1109/TPAMI.2012.213`.

[44] Yangming Shi, Xiaopo Wu, and Ming Zhu. "Low-light Image Enhancement Algorithm Based on Retinex and Generative Adversarial Network". In: *ArXiv* abs/1906.06027 (2019).

[45] Cheng Zhang, Qingsen Yan, Yu Zhu, Xianjun Li, Jinqiu Sun, and Yanning Zhang. "Attention-based network for low-light image enhancement". In: *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2020, pp. 1–6.

[46] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. "Non-local Neural Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7794–7803. DOI: `10.1109/CVPR.2018.00813`.

[47] Jie Hu, Li Shen, and Gang Sun. "Squeeze-and-Excitation Networks". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7132–7141. DOI: `10.1109/CVPR.2018.00745`.

[48] Dokyeong Kwon, Guisik Kim, and Junseok Kwon. "DALE : Dark Region-Aware Low-light Image Enhancement". In: *ArXiv* abs/2008.12493 (2020).

[49] Zhuqing Jiang, Chang Liu, Ya'nan Wang, Kai Li, Aidong Men, Haiying Wang, and Haiyong Luo. "Bridge the Vision Gap from Field to Command: A Deep Learning Network Enhancing Illumination and Details". In: *ArXiv* abs/2101.08039 (2021).

[50] P. Huber. "Robust Estimation of a Location Parameter". In: *Annals of Mathematical Statistics* 35 (1964), pp. 492–518.

[51] Chongyi Li, Chunle Guo, and Chen Change Loy. "Learning to Enhance Low-Light Image via Zero-Reference Deep Curve Estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). DOI: `10.1109/TPAMI.2021.3063604`.

[52] Sungkwon An, Hyungmin Roh, and Myungjoo Kang. "Blur Invariant Kernel-Adaptive Network for Single Image Blind deblurring". In: *arXiv: Computer Vision and Pattern Recognition* (2020).

[53] John Immerkær. "Fast Noise Variance Estimation". In: *Computer Vision and Image Understanding* 64.2 (1996), pp. 300–302. ISSN: 1077-3142. DOI: `https://doi.org/10.1006/cviu.1996.0060`. URL: `https://www.sciencedirect.com/science/article/pii/S1077314296900600`.

[54] Rémi Cogranne. "Determining JPEG image standard quality factor from the quantization tables". In: *arXiv preprint arXiv:1802.00992* (2018).

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[56] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. "Learning Enriched Features for Real Image Restoration and Enhancement". In: *ECCV*. 2020.

[57] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: `10.1109/CVPR.2017.243`.

[58] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. "Two deterministic half-quadratic regularization algorithms for computed imaging". In: *Proceedings of 1st International Conference on Image Processing*. Vol. 2. IEEE. 1994, pp. 168–172.

[59] *Pytorch: From Research To Production*. URL: `https://pytorch.org/` (visited on 05/14/2021).

[60] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: ICML'10. Haifa, Israel: Omnipress, 2010, pp. 807–814. ISBN: 9781605589077.

[61] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. "The 2017 DAVIS Challenge on Video Object Segmentation". In: *arXiv:1704.00675* (2017).

[62] Diederik Kingma and Jimmy Ba. "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations* (Dec. 2014).

[63] Leon B Lucy. "An iterative technique for the rectification of observed distributions". In: *The astronomical journal* 79 (1974), p. 745.

[64] William Hadley Richardson. "Bayesian-based iterative method of image restoration". In: *JoSA* 62.1 (1972), pp. 55–59.

[65] Simone Zini, Simone Bianco, and Raimondo Schettini. "Deep Residual Autoencoder for Blind Universal JPEG Restoration". In: *IEEE Access* 8 (2020), pp. 63283–63294. DOI: 10.1109/ACCESS.2020.2984387.

[66] Pablo Arbeláez, Michael Maire, Charless Fowlkes, and Jitendra Malik. "Contour Detection and Hierarchical Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), pp. 898–916. DOI: 10.1109/TPAMI.2010.161.

[67] Xiaoshuai Zhang, Wenhan Yang, Yueyu Hu, and Jiaying Liu. "Dmcnn: Dual-Domain Multi-Scale Convolutional Neural Network for Compression Artifacts Removal". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018, pp. 390–394. DOI: 10.1109/ICIP.2018.8451694.

[68] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. "Multi-level wavelet-CNN for image restoration". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 773–782.

[69] Bolun Zheng, Rui Sun, Xiang Tian, and Yaowu Chen. "S-Net: a scalable convolutional neural network for JPEG compression artifact reduction". In: *Journal of Electronic Imaging* 27 (2018).

[70] Leonardo Galteri, Lorenzo Seidenari, Marco Bertini, and Alberto Del Bimbo. "Deep Universal Generative Adversarial Compression Artifact Removal". In: *IEEE Transactions on Multimedia* 21.8 (2019), pp. 2131–2145. DOI: 10.1109/TMM.2019.2895280.